Fraud Detection in New York City Property

Project 1 Report

DSO 562 - Fraud Analytics
February 2018

TEAM 2

Louis Yansaud | Olivia Huang | Jamie Lee | Tong Xie | Conglin Xu | Ravi Gangumalla

TABLE OF CONTENTS

Table Of Contents.....2

Executive Summary.....3
Project Description.....3
Project Goal.....3
Key Findings.....3

Data Understanding.....4
Description of the Data.....4
Numerical Variables.....4
Categorical Variables.....5

Data Cleaning.....6
Removing Variables.....6
Filling in Missing Values.....7
Feature Engineering.....8

Data Modeling.....12
Fraud Algorithm.....12
Calculation.....14
PCA.....16
AUTOENCODER.....21

Data Evaluation.....23
Comparison of the Two Algorithmic Methods.....23

Conclusion.....28
Results.....28
Fraud Scenarios.....30
Potential Improvement.....30

Appendix.....31
Data Quality Report.....31

EXECUTIVE SUMMARY

PROJECT DESCRIPTION

This report provides a detailed analysis of 'New York City Property Tax Valuation Data' and also aims at identifying potential fraudulent records from the data set using unsupervised machine learning methods.

The programming tools used for Data Cleaning were performed on Microsoft Excel and R. The original dataset contains unique records of more than 1 million properties across the state of New York with 30 different fields, both numerical and categorical. Necessary feature analysis include:

- Data Understanding and Data Cleaning (estimating missing values of each fields)
- Data Modeling (creating expert variables) and Data Evaluation
- Fraud Score Calculation (Heuristic Algorithm and Autoencoder)

PROJECT GOAL

Our objective for this project is to use machine learning algorithms such as Autoencoder and Heuristic Algorithms and calculate a fraud score for each of the New York Property data records to analyze anomalies for potential fraud detection. By applying both of the unsupervised machine learning methods, records with high scores show to be potentially fraudulent. The report will explain each step in complete detail.

KEY FINDINGS

The two fraud detection algorithms we used, Heuristic and Autoencoder-Based Anomaly Detection, had a considerably high overlap matching percentage among the top 1% of all records:

- Missing values were properly filled using reasonable data cleaning methodology.
- The 51 expert variables were carefully crafted to perform PCA analysis.
- After normalizing the fraud scores and visualizing the scores, we found both score distributions to be right skewed.
- Among the top 10 highest fraud score records, anomalies were found in several fields.

DATA UNDERSTANDING

DESCRIPTION OF THE DATA

The dataset originates from a public source of the New York City Department of Finance (DOF). This dataset contains records of every available property in New York City. It was first created in September 2011, and was most recently updated in December 2017. For the purpose of this project, we have selected a dataset version from November 2011. The dataset contains 1,048,575 observations with 30 variables.

Below is a list of variables, categorized by numerical, categorical and other, with short descriptions and percent populated values.

NUMERICAL VARIABLES

Table 1. Numerical Variables

Numerical Variables	Description	Percent Populated (%)
RECORD	Unique identifier	100.0
BBLE	Concatenation of AV_BORO, AV_BLOCK, AV_LOT, AV_EASEMENT	100.0
BLOCK	Valid block ranges by BORO	100.0
LOT	Unique number within BORO/BLOCK	100.0
LTFRONT	Lot frontage in feet	100.0
LTDEPTH	Lot depth in feet	100.0
STORIES	The story number of the building	95.0
FULLVAL	Total market value	100.0
AVLAND	Assessed value for the land	100.0
AVTOT	Assessed value for total market	100.0
EXLAND	Exempt value for the land	100.0
EXTOT	Exempt value for total market	100.0
EXCD1	Exempt class code	59.4

BLDFRONT	Building frontage in feet	100.0
BLDEPTH	Building depth in feet	100.0
AVLAND2	Another version of assessed value for the land	26.8
AVTOT2	Another version of assessed value for total market	26.8
EXLAND2	Another version of exempt value for the land	8.3
EXTOT2	Another version of exempt value for total market	12.4
EXCD2	Another version of exempt class code	8.7

CATEGORICAL VARIABLES

Table 2. Categorical Variables

Categorical Variables	Description	Percent Populated (%)
EASEMENT	Be used to describe easement,12 categories	100.0
BLDGCL	Building class, 200 categories	100.0
TAXCLASS	Property tax class code (NYS Classification), 11 categories	100.0
EXMPTCL	Exempt class used for fully exempt properties only,14 categories	1.4
PERIOD	Change period of the file, only 1 category	100.0
VALTYPE	Only 1 category, 'AC-TR' means actual and transitional	100.0

OTHER VARIABLES (STRING AND DATE)

Table 3. Other Variables

Other Variables	Description	Percent Populated (%)
OWNER	Owner name of the property	97.0
STADDR	Standard address of the property	99.9

ZIP	ZIP code of the property	97.5
YEAR	Updated date of the record, only 1 value	100.0

DATA CLEANING

Our primary goals in the data cleaning process were to:

1. Consolidate the data to relevant and useful fields
2. Fill in missing records with reasonable and innocuous values

In the process, we created several new field names.

REMOVING VARIABLES

In preparation for data cleaning, we removed the following fields:

BBLE, BLOCK, LOT, OWNER, STADDR, AVLAND2, AVTOT2, EXLAND2, EXTOT2, EXCD2, EXEMPTCL, PERIOD, YEAR, and VALTYPE.

- **BBLE** is a combination of the BORO, BLOCK, LOT, and EASEMENT fields. Considering that BLOCK ranges are set by BORO code and LOT and BBLE are both unique numerical values, **BLOCK, LOT, and BBLE** are not valuable for the purposes of our data analysis and can be removed.
    - In place, we extracted the **BORO** code from BBLE and incorporated it as a new field variable for better, more insightful analysis. We kept EASEMENT as there are only 9 distinct values for this field.
    - **PERIOD, YEAR, and VALTYPE** have no missing values and hold the same values across all observations. Therefore, they should all be deleted.

- **STADDR** are street addresses and missing values in the dataset indicate missing addresses. We only need to keep ZIP code as this is the most useful geographical field.

- **OWNER** is not a numerical field. While we can clean OWNER into a specific format such as 'First Name and Last Name', it is difficult to accurately fill in missing values for owner. For this reason, OWNER is not as informative and should be deleted.

- **EXEMPTCL** does not show any interesting information for fraud and has 1,033,583 missing values so it should be deleted.

- **EXLAND2, EXTOT2, EXCD2** have low percent populated values in comparison to EXLAND, EXTOT, EXCD1. EXLAND, EXTOT, and EXCD1 are more practical and meaningful fields to use instead.

FILLING IN MISSING VALUES

- We discovered that **EASEMENT** values of F, H, I,...M are actually duplicates of E and empty observations represent “No Easement” rather than missing values. To simplify the data, we replaced all values F through M to E and filled in empty records with “NO”. The resulting EASEMENT is now a categorical field with only 9 distinct values, A, B, E, N, P, R, S, U, and NO.

- **ZIP** has a total of 26,356 missing values. Of these missing values, there are two scenarios:
  (1) Has both a state address, in addition to BORO
  (2) Has no associated state address and only BORO

  Our process to match the correct missing ZIP to each property was as follows:

  - The STADDR field is completely populated, meaning every property has a BORO value associated with it. We found that the best approach to filling in the missing ZIP values is to use these BORO values. We created a new field called STADDRBORO to match each state address with its BORO code value within one condensed field, i.e. “98 ELTON STREET, Brooklyn”.

  - For case (1), missing ZIP values with BORO and a state address, we used the ggmap package in R to download data from Google Maps API and obtained longitude and latitude values for each address. We created a geospatial function to apply on each of these longitude and latitude values and extract the correct ZIP codes. However, because Google Maps geocoding API has a 2,500 query limit and time allocation limit per user each day, we were only able to replace 2,500 of the missing values. For the remaining values, we had to come up with an alternative method.
  - We decided to use the most common ZIP code within each BORO to fill in the rest of the missing values. For example, in Manhattan, the most common ZIP is 10023 so we filled all the missing values with no state address in Manhattan with ZIP 10023.

  - For case (2), missing ZIP values with no state address, we used the same methodology and filled in these with the most common ZIP according to which BORO the property is located in.

- To fill in missing values for **STORIES**, we grouped by different variables, including ZIP, EASEMENT, TAXCLASS, and BLDCL, and calculated the averages. We found the range of averages to be the following, when grouping by:
  - ZIP: 1 to 66.3
  - EASEMENT: 1 to 6
  - **TAXCLASS 1 to 16**
  - BLDGCL: 1 to 36
  - We chose to group by TAXCLASS because the range of averages was not too small and still has variation. We filled in the missing values according to which tax class they belonged to, taking the average STORIES value from that tax class.

- For **LTFRONT and LTDEPTH**, we first grouped by different variables and found the following range for averages:
    - BLDGCL: 20.2 to 3550
    - ZIP: 15 to 2000
    - TAXCLASS: 25.9 to 338
    - EASEMENT: 12.5 to 259
    - **BORO (for LTFRONT): 33.45 to 81.43**
    - **BORO (for LTDEPTH): 101 to 120**
    - The range for averages grouped by BORO is the best choice here. Since LTDEPTH and LTFRONT have the same range, we decided to group both by BORO. We filled in missing values according to the average value associated with the BORO of the given record.

- For the purposes of staying consistent, we replaced missing values for **BLDFRONT and BLDDEPTH** by grouping by BORO as well.

- For **FULLVAL, AVTOT, and AVLAND**, we grouped by ZIP because within the same ZIP code area, we usually have similar property values. We first checked the range for averages when grouping by ZIP:
    - FULLVAL: 197,248.2 to 378,000,000
    - AVTOT: 25,423.67 to 170,100,000.00
    - AVLAND: 10,524.36 to 47,700,000
    - We replaced the missing values based on their ZIP.

FEATURE ENGINEERING

We first created 3 sizes for scaling per experts' advice:

- **LOTAREA** = LTFRONT * LTDEPTH
- **BLDAREA** = BLDFRONT * BLDDEPTH
- **BLDVOL** = BLDAREA * STORIES

Then we calculated 9 variables by normalizing each of the 3 expert variables (**FULLVAL, AVLAND, AVTOT**) by the 3 sizes above.

- FULLVAL/BLDAREA
- FULLVAL/BLDVOL
- AVLAND/LOTAREA
- AVLAND/BLDAREA
- AVLAND/BLDVOL
- AVTOT/LOTAREA
- AVTOT/BLDAREA
- AVTOT/BLDVOL

Since **ZIP**, **TAXCLASS** and **BORO** have influence in the value of the property, we created 45 variables for the averages of the above 9 variables grouped by **ZIP5** (taking first five digits of ZIP), **ZIP3** (taking first three digits of ZIP), **TAXCLASS**, **BORO** and **ALL** (the overall mean).

Additionally, we created 6 more variables by normalizing the FULLVAL, AVLAND and AVTOT with **ZIP** and **BORO**. For example, $\frac{FULLVAL}{FULLVAL\ MEAN_{ZIP}}$ is derived with FULLVAL which falls into a specific ZIP divided by the mean of FULLVAL grouped by this ZIP.

Following is a list of expert variables we created. In total, we created 51 expert variables.

Table 4. List of Expert Variables

Expert Variables	Calculation
1	FULLVAL_BLDAREA = $\frac{FULLVAL}{BLDAREA}$, $\frac{FULLVAL\_BLDAREA}{FULLVAL\_BLDAREA\ MEAN_{ZIP}}$
2	FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{ZIP}}$
3	FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{ZIP}}$
4	AVTOT_BLDAREA = $\frac{AVTOT}{BLDAREA}$, $\frac{AVTOT\_BLDAREA}{AVTOT\_BLDAREA\ MEAN_{ZIP}}$
5	AVTOT_BLDVOL = $\frac{AVTOT}{BLDVOL}$, $\frac{AVTOT\_BLDVOL}{AVTOT\_BLDVOL\ MEAN_{ZIP}}$
6	AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{ZIP}}$
7	AVLAND_BLDAREA = $\frac{AVLAND}{BLDAREA}$, $\frac{AVLAND\_BLDAREA}{AVLAND\_BLDAREA\ MEAN_{ZIP}}$
8	AVLAND_BLDVOL = $\frac{AVLAND}{BLDVOL}$, $\frac{AVLAND\_BLDVOL}{AVLAND\_BLDVOL\ MEAN_{ZIP}}$
9	AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{ZIP}}$
10	FULLVAL_BLDAREA = $\frac{FULLVAL}{BLDAREA}$, $\frac{FULLVAL\_BLDAREA}{FULLVAL\_BLDAREA\ MEAN_{ZIP3}}$
11	FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{ZIP3}}$
12	FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{ZIP3}}$

13	AVTOT_BLDAREA = $\frac{AVTOT}{BLDAREA}$, $\frac{AVTOT\_BLDAREA}{AVTOT\_BLDAREA\ MEAN_{ZIP3}}$
14	AVTOT_BLDVOL = $\frac{AVTOT}{BLDVOL}$, $\frac{AVTOT\_BLDVOL}{AVTOT\_BLDVOL\ MEAN_{ZIP3}}$
15	AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{ZIP3}}$
16	AVLAND_BLDAREA = $\frac{AVLAND}{BLDAREA}$, $\frac{AVLAND\_BLDAREA}{AVLAND\_BLDAREA\ MEAN_{ZIP3}}$
17	AVLAND_BLDVOL = $\frac{AVLAND}{BLDVOL}$, $\frac{AVLAND\_BLDVOL}{AVLAND\_BLDVOL\ MEAN_{ZIP3}}$
18	AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{ZIP3}}$
19	FULLVAL_BLDAREA = $\frac{FULLVAL}{BLDAREA}$, $\frac{FULLVAL\_BLDAREA}{FULLVAL\_BLDAREA\ MEAN_{TAXCLASS}}$
20	FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{TAXCLASS}}$
21	FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{TAXCLASS}}$
22	AVTOT_BLDAREA = $\frac{AVTOT}{BLDAREA}$, $\frac{AVTOT\_BLDAREA}{AVTOT\_BLDAREA\ MEAN_{TAXCLASS}}$
23	AVTOT_BLDVOL = $\frac{AVTOT}{BLDVOL}$, $\frac{AVTOT\_BLDVOL}{AVTOT\_BLDVOL\ MEAN_{TAXCLASS}}$
24	AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{TAXCLASS}}$
25	AVLAND_BLDAREA = $\frac{AVLAND}{BLDAREA}$, $\frac{AVLAND\_BLDAREA}{AVLAND\_BLDAREA\ MEAN_{TAXCLASS}}$
26	AVLAND_BLDVOL = $\frac{AVLAND}{BLDVOL}$, $\frac{AVLAND\_BLDVOL}{AVLAND\_BLDVOL\ MEAN_{TAXCLASS}}$
27	AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{TAXCLASS}}$
28	FULLVAL_BLDAREA = $\frac{FULLVAL}{BLDAREA}$, $\frac{FULLVAL\_BLDAREA}{FULLVAL\_BLDAREA\ MEAN_{BORO}}$
29	FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{BORO}}$
30	FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{BORO}}$
31	AVTOT_BLDAREA = $\frac{AVTOT}{BLDAREA}$, $\frac{AVTOT\_BLDAREA}{AVTOT\_BLDAREA\ MEAN_{BORO}}$

32	AVTOT_BLDVOL = $\frac{AVTOT}{BLDVOL}$, $\frac{AVTOT\_BLDVOL}{AVTOT\_BLDVOL\ MEAN_{BORO}}$
33	AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{BORO}}$
34	AVLAND_BLDAREA = $\frac{AVLAND}{BLDAREA}$, $\frac{AVLAND\_BLDAREA}{AVLAND\_BLDAREA\ MEAN_{BORO}}$
35	AVLAND_BLDVOL = $\frac{AVLAND}{BLDVOL}$, $\frac{AVLAND\_BLDVOL}{AVLAND\_BLDVOL\ MEAN_{BORO}}$
36	AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{BORO}}$
37	FULLVAL_BLDAREA = $\frac{FULLVAL}{BLDAREA}$, $\frac{FULLVAL\_BLDAREA}{FULLVAL\_BLDAREA\ MEAN_{ALL}}$
38	FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{ALL}}$
39	FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{ALL}}$
40	AVTOT_BLDAREA = $\frac{AVTOT}{BLDAREA}$, $\frac{AVTOT\_BLDAREA}{AVTOT\_BLDAREA\ MEAN_{ALL}}$
41	AVTOT_BLDVOL = $\frac{AVTOT}{BLDVOL}$, $\frac{AVTOT\_BLDVOL}{AVTOT\_BLDVOL\ MEAN_{ALL}}$
42	AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{ALL}}$
43	AVLAND_BLDAREA = $\frac{AVLAND}{BLDAREA}$, $\frac{AVLAND\_BLDAREA}{AVLAND\_BLDAREA\ MEAN_{ALL}}$
44	AVLAND_BLDVOL = $\frac{AVLAND}{BLDVOL}$, $\frac{AVLAND\_BLDVOL}{AVLAND\_BLDVOL\ MEAN_{ALL}}$
45	AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{ALL}}$
46	FULLVAL_MEANBYZIP = $\frac{FULLVAL}{FULLVAL\ MEAN_{ZIP}}$
47	AVTOT_MEANBYZIP = $\frac{AVTOT}{AVTOT\ MEAN_{ZIP}}$
48	AVLAND_MEANBYZIP = $\frac{AVLAND}{AVLAND\ MEAN_{ZIP}}$
49	FULLVAL_MEANBYBORO = $\frac{FULLVAL}{FULLVAL\ MEAN_{BORO}}$
50	AVTOT_MEANBYBORO = $\frac{AVTOT}{AVTOT\ MEAN_{BORO}}$

51	AVLAND_MEANBYBORO = $\frac{AV\,LAND}{AV\,LAND\;\;MEAN_{\;BORO}}$

DATA MODELING

After creating a total of 51 expert variables, we used the Principal Component Analysis (PCA) method to transform the New York Property dataset of high-dimensional data. Because PCA can be applied only on numerical data, we created a new dataset that includes only the 51 expert variables.

Principal Component Analysis is a normalized linear combination of the original predictors in a dataset. The principal components are supplied with normalized version of original predictors. This is because the original predictors may have different scales. Performing PCA on un-normalized variables may lead to insanely large loadings for variables with high variance.

In order to avoid dependency of a principal component on the variable with high variance, we then perform a Z scaling method onto the PCAs. The main purposes of a principal component analysis are to identify patterns through data analysis and to reduce the dimensions of the dataset with minimal loss of information.

Suppose we have a set of predictors as $X^1, X^2, ..., X^p$. The principal component can be written as:

$$Z^1 = \phi^{11}X^1 + \phi^{21}X^2 + \phi^{31}X^3 + ... + \phi^{p1}X^p \qquad (1.1)$$

- Z^1 is the first principal component.
- ϕ^{p1} is the loading vector comprising of loadings ($\phi^1, \phi^2, ...$) of first principal component.
- $X^1, X^2, ..., X^p$ are normalized predictors. Normalized predictors have mean equal to zero and standard deviation equal to one.

FRAUD ALGORITHM

PCA Application Process

In order to construct the fraud algorithm, the following data manipulation steps were necessary:

1. Z Scaling to prepare for feature selection/dimensionality reduction.
2. Reducing dimensions via PCA in order to summarize each variations of the New York Property dataset.
3. Z Scaling the reduced variables.

PCA Mathematical Model

1. Original matrix – dataset X (n x m), n objects, m variables:

$$X = [x_{ij}] = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{vmatrix}$$

2. $Z = [z_{ij}]$ standardized matrix X with i = 1,…., n j = 1,…., m

3. PCA aims is finding out the transformation matrix Q, which convert *m* standardized variables (matrix Z) into *m* mutual independent component (matrix P).

$$P = Z.Q$$

$$PC1 = \sum_i X_i * Z_i$$

$$PC2 = \sum_i X_i * Z_i$$

$$\vdots$$

$$PCn = \sum_i X_i * Z_i$$

4. Then we get the PCA matrix via reduction:

$$P = [p_{ij}] = \begin{vmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nm} \end{vmatrix}$$

5. Modification of $P = Z.Q$ → we get matrix Λ

$$\Lambda = Q^T \cdot R \cdot Q$$

CALCULATION

Heuristic Score - Outlier Detection Algorithm

After performing PCA, scaling and reducing the dimensionality, we combined the generated z-scores with the following heuristic algorithm that calculates the fraud score for each record i:

$$s_i = \left(\sum_k \left| z_k^i \right|^m \right)^{1/m} , \; m \; anything \qquad (1.2)$$

This equation detects anomalies by calculating the Mahalanobis Distance, a generalized distance of how far a point is from the center of the dataset. We used the absolute value to represent all the positive distances between the normalized z-scaled PC scores, z_k^i, and the origin of the principal component space. The further away the point is from the origin, the greater the distance the z-scaled PC values are from zero and thus, the higher the fraud score.

The Mahalanobis Distance is a useful metric in the context of fraud detection because we can compute the distance of each data point from the center of the dataset it is comprised in, normalized by the standard deviation of each of the dimensions and adjusted for the covariances of those dimensions. Records with a high Mahalanobis distance value are considered extreme values and are outliers that are potentially fraudulent records to focus on.

Fraud Score - Autoencoder-Based Anomaly Detection Algorithm

The reproduction error of the autoencoder is a measure of fraud score. The autoencoder takes the original z-scaled PC records, puts them through a non-linear transformation, and attempts to reproduce the exact same record.

Our algorithm takes both the original z-scaled pc records and autoencoder reproduced z-scaled records, and calculates the difference between the two values.

We used the following function to calculate the difference between the z-scaled records:

$$s_i = \left(\sum_k \left| z_k'^i - z_k^i \right|^m \right)^{1/m} , m \; anything \; , for \; each \; record \; i \qquad (1.3)$$

The functional relationship represented here is a measure of how closely the autoencoder was able to reconstruct the original inputs. For each record i, $z_k'^i$ represents the autoencoder reproduced z-score, and z_k^i is the original z-scaled PC score. The function output, s_i, represents the summation of the differences between the outputs and inputs and can be defined as the reconstruction error. Using this algorithm, we can identify the data records with high reconstruction errors and high RMSE. The data

records with high values for s_i have high reconstruction and are labelled as anomalies. This algorithm acts as an anomaly detection method because these observations with high mean squared error are the potentially the fraudulent data we are looking for.

Application to New York Property Fraud Detection

We calculated the fraud scores under both the Heuristic and Autoencoder algorithms for all records in the New York Property dataset (i = 1,...,1048575).

We then scaled the scores to range [0,1] because some field values, such as LTFRONT and LTDEPTH, both consisted of extremely and unusually small values, for example, values of 1, 2, and 3. We performed scaling using the following formula:

Let k_i denote the rescaled i^{th} fraud score resulting from the Heuristic algorithm, such that

$$k_i = \frac{s_i - min(s_1,.., s_{1048575})}{max(s_1,.., s_{1048575} - min(s_1,.., s_{1048575})} \quad , \; i = 1, ..., 1048575$$

Similarly, let a_i denote the rescaled i^{th} fraud score resulting from the Autoencoder algorithm, such that

$$a_i = \frac{s_i - min(s_1,.., s_{1048575})}{max(s_1,.., s_{1048575} - min(s_1,.., s_{1048575})} \quad , \; i = 1, ..., 1048575$$

This resulted in two separate tables corresponding to each of the two algorithms:

- Heuristic Algorithm Fraud Score Table, with fraud scores $k_1, ..., k_{1048575}$
- Autoencoder-Based Anomaly Detection Algorithm Fraud Score Table, with fraud scores $a_1, ..., a_{1048575}$

In order to match the fraud scores to the original dataset, we created a new field called "Row_Number_ID". This field corresponds to the fraud score for the n^{th} observation of the original New York Property dataset.

We then sorted both the Heuristic and Autoencoder fraud score tables by descending order of fraud score. Using the sorted tables, we found the top 1% highest fraud scores and calculated the overlap between two separate tables. We also took the top 10 records from both tables and performed further analysis, such as exploring anomalies.

PCA (Principal Component Analysis)

Principal Component Analysis (PCA) is a statistical procedure used to reduce dimensionality and remove correlations. It calculates the direction in which the variance is maximal and repeats this for the next orthonormal axis. PCA finds the dominant directions in the data, creating new variables that are linear combination of the original variables. The new variables are called “principal components” and they are ordered by the variance. Their magnitude are their eigenvalues, which are proportional to the variance in the direction of that principal component.

PCA is applied in this case since it is simple and efficient. It requires no special assumptions on the data and PCA can be all datasets. Also, PCA can deal with large datasets both in objects and variables.

Application to New York Property Fraud Detection

1. One of our primary goals in performing PCA is to detect anomalies. In the case of fraud detection, an anomaly is an observation that is significantly different from and does not conform to the expected pattern or behavior of the remaining data observations.

   By performing PCA, we generate low dimensional representations of the data and reconstruct these representations in the original data space. These reconstructions aim to represent the data in its true nature, without noise. Through PCA, we can find the difference between the original data point and its low dimensional reconstruction. We call this the reconstruction error. As mentioned above, the algorithm we used calculates this error value as the Mahalanobis distance. High reconstruction values denote a higher fraud score, anomalies within the dataset, and therefore potentially fraudulent records.

2. Another important goal in using PCA is to gain insight on which PCs explain most of the variation from the data, as well as decide which and how many PCs to retain after PCA. By performing PCA, we were able to analyze the eigenvalues and the corresponding proportions of variances.

   The eigenvalues measure how much variation is retained by each principal component. After running PCA, we used the factoextra package to plot a Scree Plot to better visualize the cumulative variance percentages represented by the top ten principal components.

Our Procedure

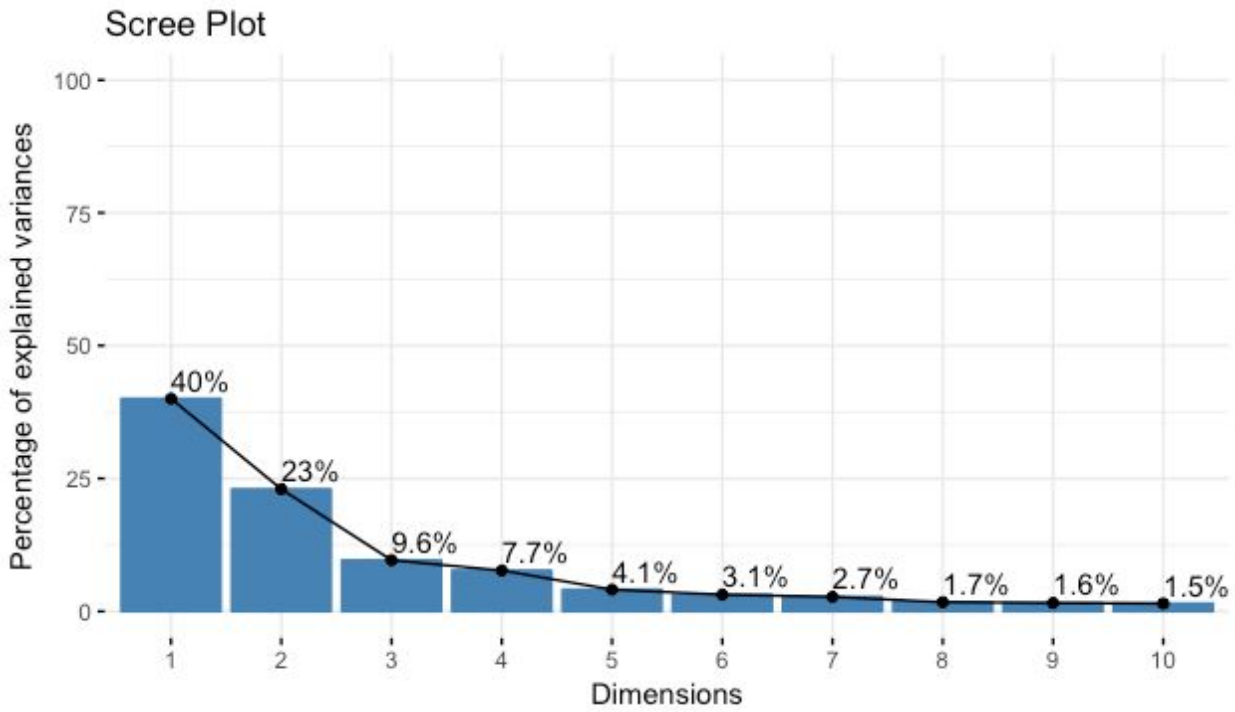
We used a built-in Principal Components Analysis function in R called "prcomp()" to perform PCA using our expert variables.

This function has two important measures, $center$ and $scale$, which correspond to the respective mean and standard deviation of the variables. By default, prcomp() centers the $mean = 0$. We set parameter $scale. = T$ to normalize the variables such that they have standard deviation equal to 1.

Using the prcomp() function, we were able to get the PCA matrix with the principal component score vectors in 1,048,575 X 51 dimension.

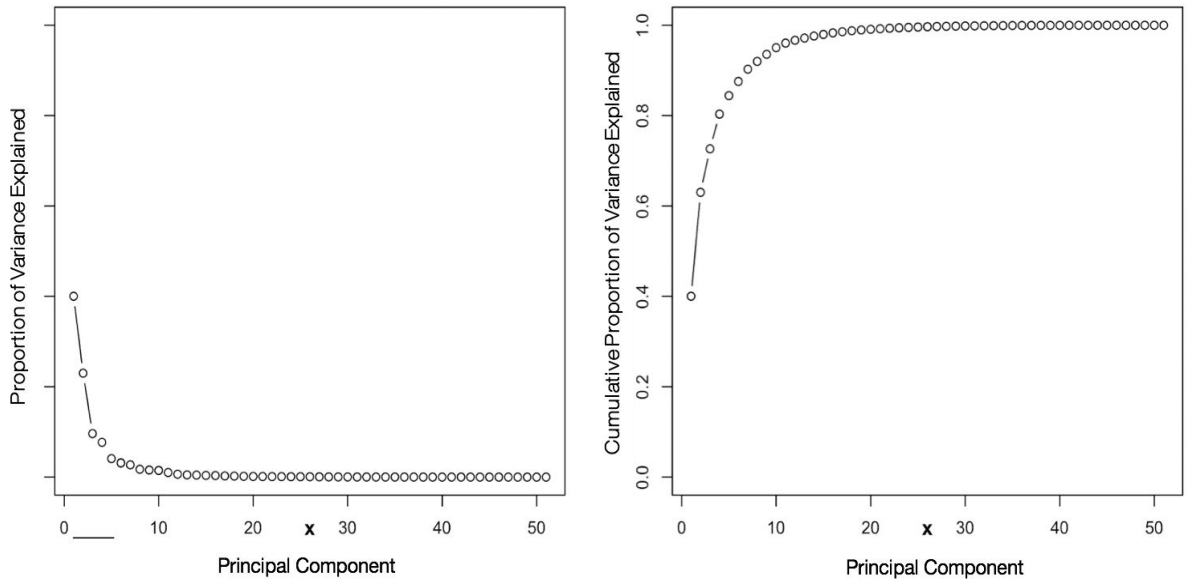
In order to decide on which PCs to keep for the fraud score calculation and data modeling, we explored which PCs explain the most variability in the data and kept the highest contributors. To do this, we first used the "fviz_eig" function from the factoextra package to plot the Scree Plot, shown in Figure 1.

Figure 1. Scree Plot of PCs and Percentage of Explained Variances



This plot shows the cumulative variances explained by each of the top ten PCs, labelled as dimensions 1 through 10. We found that the first three principal components explain 72.6% of the variances contained in the data.

Figure 2. Proportion (Left) and Cumulative (Right) Proportion of Variances



In Figure 2., we used a pcaCharts() function in R to plot the overall summary of proportions of variances explained by all 51 PCs. The figure on the right depicts cumulative variance. By PC10, nearly all the variability in the data is represented. For our analysis, we chose to keep the top 8 PCs, which cumulatively retain 91.9% of the variances in the data. On these top 8 PCs, we performed z-scaling once again.

That is, for each of the PCs, we z-scaled using the following formula:

$$z_i = \frac{PC_i - \mu_{PC_i}}{\sigma_{PC_i}}, \; i = 1, \ldots, 8$$

We computed $z_1, z_2, z_3, z_4, z_4, z_5, z_6, z_7, z_8$, the new z-scaled PC records for the top 8 PCs.

With these 8 z-scores, we calculated the fraud scores using the heuristic algorithm.

For further exploration on the expert variables, we used the corrplot and factoextra packages to plot quality of representation graphs, Figures 3 and 4. In both, we use cos2 value as a measure for the importance of that component for the given observation. Figure 3 shows the representation of all 51 expert variables across all 51 PCs. Darker blue dots mean that within that PC, that variable has a high cos2 value, which is a measure of quality of representation. For instance, for PC1, Variables 4, 22 and 28 have higher quality of representation within that PC.

Figure 3. Cos2 of Expert Variables on All Dimensions

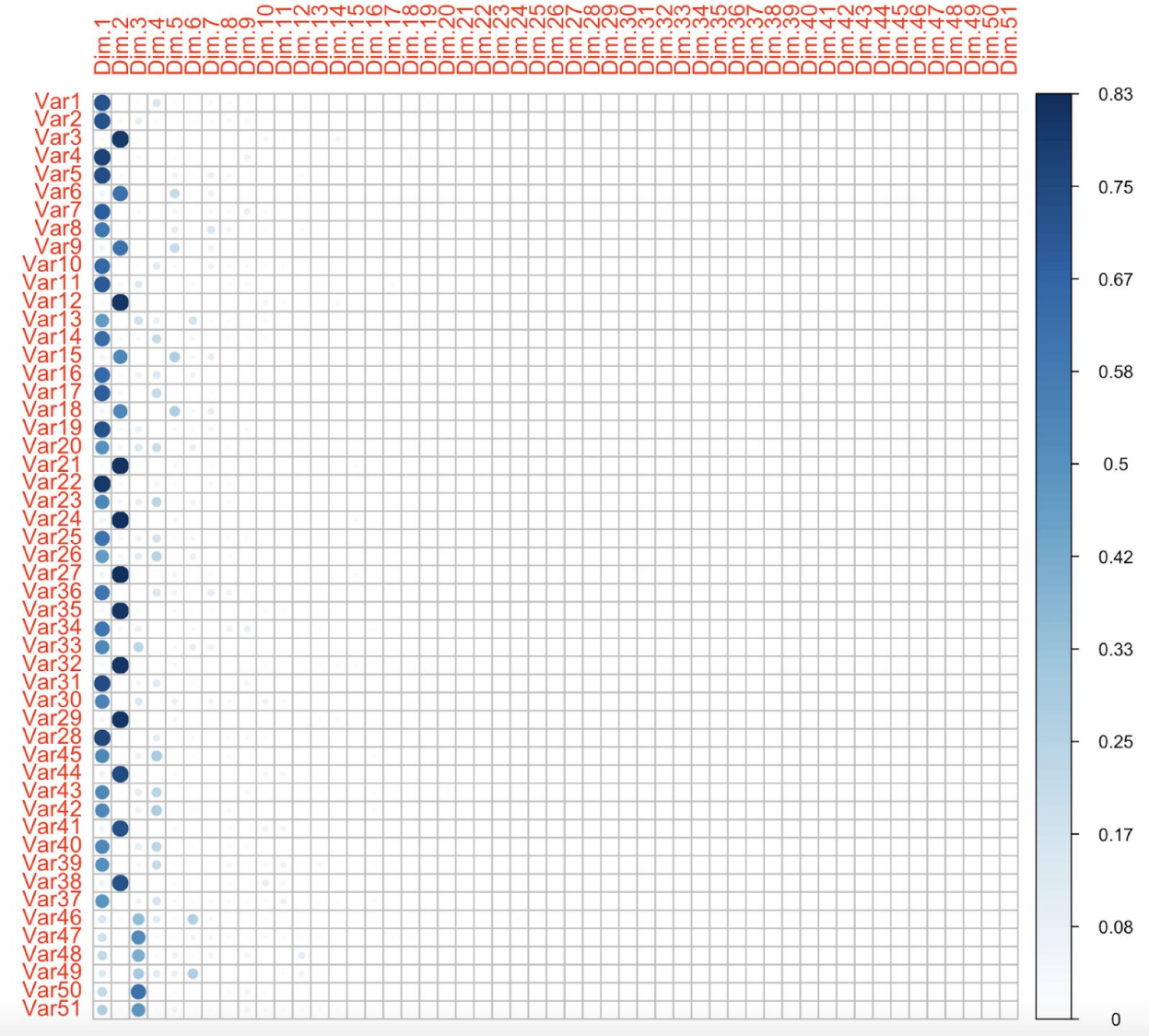


Figure 4. Cos2 of Variables on PC1 and PC2

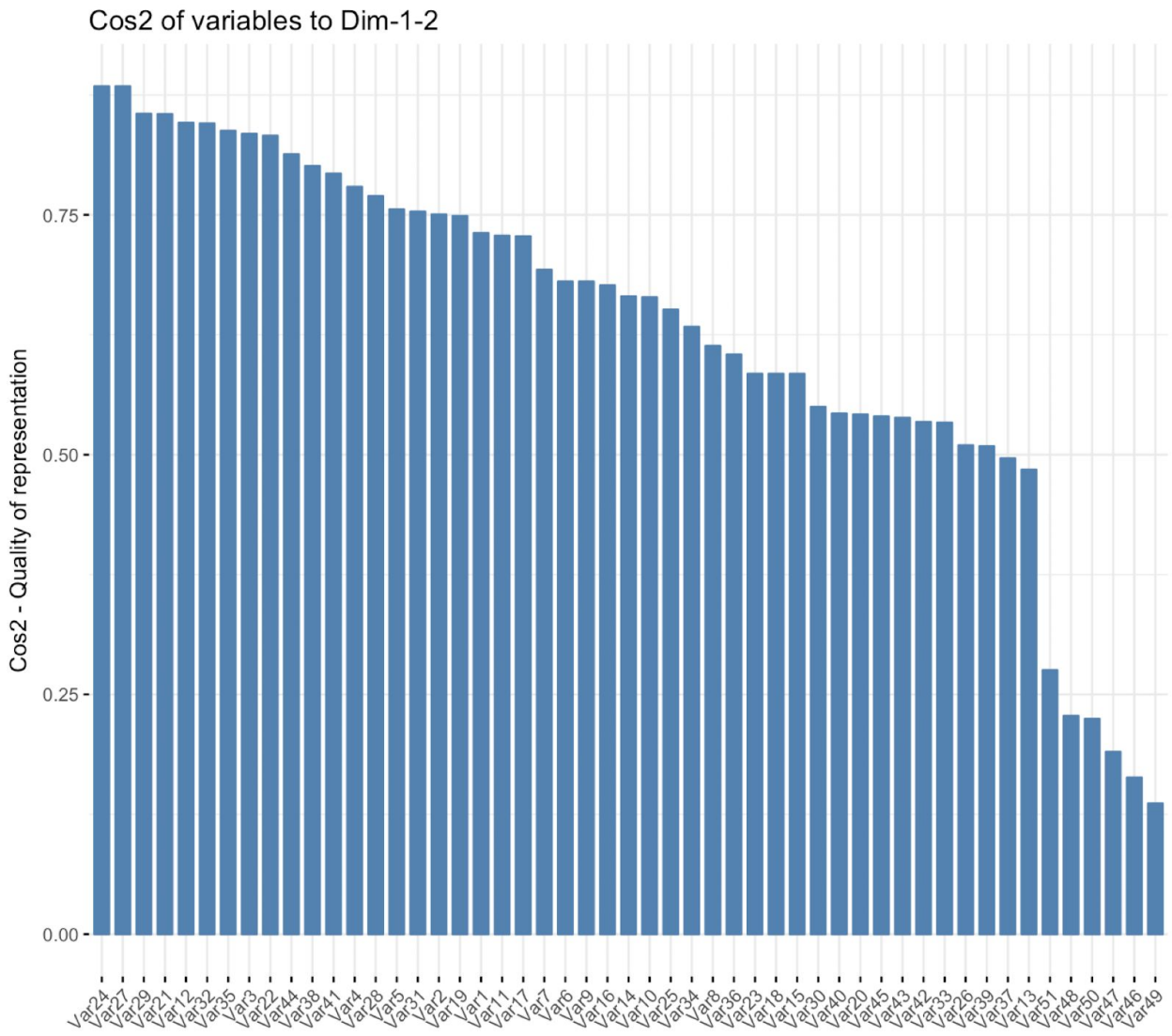


Figure 4 visualizes the top contributing variables to the first two principal components, PC1 and PC2. We can easily see that Variables 24 and 27 contribute the most to PC1 and PC2. Following these, Variables 29 and 21 have high values of contribution and are top contributors to PC1 and PC2.

Here are the list of the top contributors created on R:

Var24: AVTOT_LOTAREA = $\frac{AVTOT}{LOTAREA}$, $\frac{AVTOT\_LOTAREA}{AVTOT\_LOTAREA\ MEAN_{TAXCLASS}}$

Var27: AVLAND_LOTAREA = $\frac{AVLAND}{LOTAREA}$, $\frac{AVLAND\_LOTAREA}{AVLAND\_LOTAREA\ MEAN_{TAXCLASS}}$

Var29: FULLVAL_BLDVOL = $\frac{FULLVAL}{BLDVOL}$, $\frac{FULLVAL\_BLDVOL}{FULLVAL\_BLDVOL\ MEAN_{BORO}}$

Var21: FULLVAL_LOTAREA = $\frac{FULLVAL}{LOTAREA}$, $\frac{FULLVAL\_LOTAREA}{FULLVAL\_LOTAREA\ MEAN_{TAXCLASS}}$

AUTOENCODER

Another unsupervised model we used is the autoencoder, which calculates the reproduction error. Since the reproduction error is a measure of the record's anomaly, it is thus a fraud score.

The autoencoder is a neural network that consists of two parts, the encoder and the decoder. The encoder is the lower part of the network and embeds the z-scaled PC records into the lower dimensional array, while the decoder takes this array and decodes it. Unlike PCA, using neural networks allows us to define non-linear and complex forms that we will be able to fit with our data.

Given the input of z-scaled PC records, the autoencoder network will learn the patterns and aims to output the same records, as closely as possible. The autoencoder is powerful for fraud detection because it learns the normal patterns within the training data and as mentioned previously, we can calculate the reconstruction error and identify high errors as potentially fraudulent points.

Application to New York Property Fraud Detection

1. Autoencoder is much more flexible than PCA, since PCA only makes sense of linear methodology.
2. The autoencoder allows us to find a low dimensional representation of the input data.
3. We can train the autoencoder to learn the features of the training data to form a deep Autoencoder network.
4. The Autoencoder network is trained by minimizing the difference between the input and output.

Our Process

We used the autoencoder package in R to implement our autoencoder. We first set up the autoencoder architecture and assigned the following autoencoder network parameters:

- We set the number of layers to 3 total: 1 input, 1 hidden, and 1 output layer $(nl = 3)$
  - We defined $Input\ Layer\ L_1,\ Hidden\ Layer\ L_2,\ Output\ Layer\ L_3$.
- We used a logistic activation function to rescale the fraud score output to be in range [0,1] $(unit.type = "logistic")$

  $f(z) = \frac{1}{1+e^{-z}}$

  Each neuron in the autoencoder neural network corresponds to the mapping, from input to output, that is defined by this logistic activation function. Given that this is a sigmoid function, this ensures that the range of the autoencoder output is [0,1]

- We trained the autoencoder on training image patches of size 10 x 10 pixels $(Nx.patch = 10,\ Ny.patch = 10)$
- The number of units in the hidden layer we set to 5 $(N.\ hidden = 5)$
- The number of units in the input layer, by definition, was set to $N.input = Nx.patch * Ny.patch$
- Weight decay parameter used for preventing overfitting by decreasing magnitude of the weights, $\lambda = 0.0002$
- Weight of sparsity penalty term used as a learning rate parameter for algorithm, $\beta = 6$
- Desired sparsity parameter used to specify our desired level of sparsity, $\rho = 0.01$
- Small parameter for initialization of autoencoder weights, $\varepsilon = 0.001$
- The number of iterations in the optimizer we set to 100 $(max.iterations = 100)$

We then used the two key functions from the autoencoder package "autoencode()" and "predict()". We trained the autoencoder with 5 hidden units on 10x10 pixel inputs. Our inputs have pixel intensity values from a 10x10 image (100 pixels) so the number of units in one input layer is 100 (N.input = 100). There are 5 hidden layers in L_2 so the autoencoder must learn a compressed version of the input based on these few layers.

We used the autoencode() function to train the autoencoder and used the following inputs for the function:

- We defined the training matrix to be the matrix of PCs resulting from PCA, called "data_pca_scores22" in our R code $(X.train = data\_pca\_scores22)$
  - Our training data consists of 10x10-pixel images
- The following parameters are defined as above: $nl,\ N.hidden,\ unit.type,\ \lambda,\ \beta,\ \varrho,\ \varepsilon,\ max.iterations$
- We used BFGS optimization method for searching the minimum of the cost function $(optim.method = "BFGS")$
- We set rescale.flag to be TRUE to uniformly rescale the training matrix so all the values of the input channels are within the range of [0,1] for 'logistic' unit outputs $(rescale.flag = TRUE)$
- We left the rescaling.offset value as defined by default to 0.001, to rescale to [offset, 1-offset] for 'logistic' units $(rescaling.offset = 0.001)$

We ran the autoencode() function taking the inputs above and during the training process, optimization is performed via the BFGS optimization function.

Following this, to predict the outputs of the autoencoder, we ran the "predict()" function. The output of this function is a matrix of all the autoencoder reproduced z-scores. The inputs for this function consist of:

- The object is the output of the autoencoder function we ran previously, defined in our R code as autoencoder.object11
  $object = autoencoder.object11$
- The matrix of inputs is the matrix of PCs resultings from PCA
  $X.input = data\_pca\_scores22$
- The hidden output is set to FALSE, to tell the function to produce outputs of units in the output layer instead of the hidden layer
  $hidden.output = FALSE$
- We set up "$X.output" to get the output matrix, with rows corresponding to outputs of units generated by the output layer.

After running the predict function, we took the matrix of autoencoder reproduced z-scores and used this to input into the Autoencoder-Based Anomaly Detection Algorithm.

DATA EVALUATION

COMPARISON OF THE TWO ALGORITHMIC METHODS

After calculating the Heuristic and Autoencoder Algorithms, we visualized the distribution by plotting a histogram for each of the two methods on R.

Figure 5. Fraud Score of Heuristic Algorithm

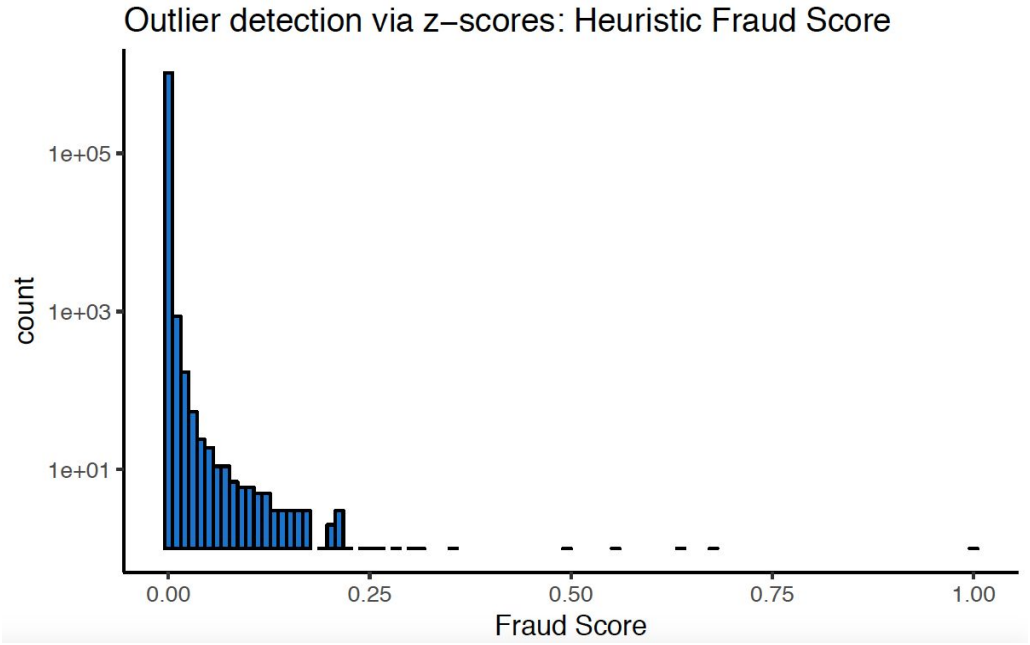
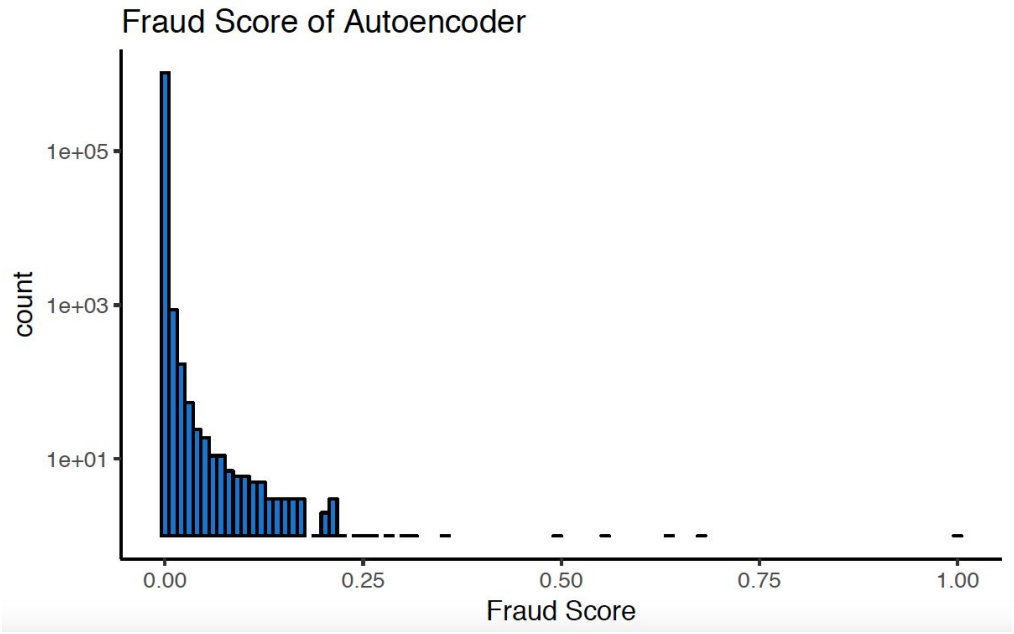


Figure 6. Fraud Score of Autoencoder



After plotting the distribution of the the two scores, we applied a Quantile binning to assign the same number of observations to each bin:

- We first sorted the records by the score in descending order then we assigned each scores into equal bins. We created 1000 bins total so that each bin will have about 1000 records in it .
- For all the records in the highest bin, we replaced the score by the number 1000. So the top, highest-scoring .1% of the records now all have the same score, 1000. For the next bin, which is the next .1% highest scoring records, we replaced their score by the number 999. The second-highest.1% scoring records now all have the same score of 999. We keep replacing each of the scores by the bin number of where they are in the sorted order.
- We used the function “ntile” from the package Dplyr in R in order to perform the Quantile Binning.

Below are the two tables showing the summary statistics of both fraud score methods from bins 1000 to bins 990.

Table 4. Fraud Score of Heuristic Algorithm from Score Bins 1000 to 990

Score Bins	Min	Q1	Mean	Median	Q3	Max	SD
1000	0.005787	0.007517	0.025304	0.010976	0.019450	1.000000	0.058835
999	0.003439	0.003785	0.004339	0.004263	0.004829	0.005781	0.000646
998	0.002538	0.002688	0.002925	0.002893	0.003143	0.003437	0.000260
997	0.002083	0.002175	0.002291	0.002283	0.002408	0.002537	0.000131
996	0.001789	0.001850	0.001927	0.001928	0.001995	0.002083	0.000085
995	0.001567	0.001616	0.001671	0.001669	0.001726	0.001788	0.000064
994	0.001393	0.001428	0.001472	0.001468	0.001513	0.001567	0.000050
993	0.001261	0.001289	0.001322	0.001319	0.001352	0.001392	0.000038
992	0.001162	0.001183	0.001209	0.001208	0.001232	0.001261	0.000028
991	0.001075	0.001096	0.001118	0.001117	0.001140	0.001161	0.000025
990	0.001002	0.001021	0.001038	0.001038	0.001058	0.001075	0.000021

Table 5. Fraud Score of Autoencoder Algorithm from Score Bins 1000 to 990

Score Bins	Min	Q1	Mean	Median	Q3	Max	SD
1000	0.005012	0.006358	0.021555	0.009348	0.016206	1.000000	0.053533
999	0.002939	0.003196	0.003736	0.003587	0.004229	0.005008	0.000606
998	0.002269	0.002383	0.002554	0.002543	0.002698	0.002939	0.000188
997	0.001826	0.001918	0.002030	0.002029	0.002133	0.002269	0.000127
996	0.001553	0.001608	0.001680	0.001679	0.001747	0.001825	0.000079
995	0.001361	0.001403	0.001451	0.001448	0.001500	0.001553	0.000056
994	0.001224	0.001254	0.001289	0.001287	0.001323	0.001361	0.000039
993	0.001108	0.001136	0.001164	0.001162	0.001191	0.001224	0.000033
992	0.001019	0.001038	0.001063	0.001063	0.001087	0.001108	0.000027
991	0.000940	0.000956	0.000978	0.000977	0.000998	0.001019	0.000023
990	0.000881	0.000896	0.000910	0.000908	0.000923	0.000940	0.000016

We then decided to analyze the overlap between the Heuristic Algorithm and the Autoencoder Algorithm results by looking at the top 1% of high score records, which is about 10,000 records. By using an inner-join on R to match the overlapped records, we found that about 80% of the records from both fraud score methods matched. Given the fraudulent record numbers outputted by both of algorithms, we cross-referenced with the original dataset to investigate every field associated with these records. The following tables present the top 10 records under each Algorithm, along with the detailed fields.

Table 5. Top 10 Fraud Score of Heuristic Algorithm

RECORD	BBLE	BORO	X	BLOCK	LOT	EASEMENT	OWNER	BLDGCL	TAXCLASS	LTFRONT	LTDEPTH
409017	4010440001E	4	Queens	1044	1	E	NYC MARINE & AVIATION	Z7	4	6	50
412707	4120130034	4	Queens	12013	34		PARKER, SHAWNESE Y	V0	1B	30	100
476422	4097780134	4	Queens	9778	134			B1	1	46	40
700160	4026460036	4	Queens	2646	36		PIOTR ZUKOWSKI	B1	1	25	119
802546	4000341080	4	Queens	34	1080			R4	2	0	0
868035	1015840020	1	Manhattan	1584	20		PRIME EAST REALTY LLC	C4	2B	25	100
980961	3009570062	3	Brooklyn	957	62		PRESIDENTIAL OWNERS C	D4	2	124	95
1015400	3012141005	3	Brooklyn	1214	1005			R1	2C	0	0
1019051	3078650023	3	Brooklyn	7865	23		YOUNGSTEIN LARRY	A1	1	30	100
1029659	4000331005	4	Queens	33	1005			R1	2C	0	0

RECORD	STORIES	FULLVAL	AVLAND	AVTOT	EXLAND	EXTOT	EXCD1	STADDR	ZIP	EXMPTCL	BLDFRONT	BLDDEPTH
409017	NA	2900	1305	1305	1305	1305	2198	24 AVENUE	NA	X1	0	0
412707	NA	195000	2638	2638	0	0	NA	142 STREET	10467		0	0
476422	2	534000	13539	24675	0	0	NA	158-20 85 AVENUE	11432		27	21
700160	2	478000	13117	23223	1620	1620	1017	56-20 62 AVENUE	11378		20	52
802546	12	163831	6198	73724	2793	70319	5113	5-49 BORDEN AVENUE	11101		0	0
868035	5	3470000	77724	212365	0	0	NA	539 EAST 87 STREET	10128		25	73
980961	4	1890000	206550	850500	45980	45980	1017	759 PRESIDENT STREET	11215		25	90
1015400	5	133212	3841	19346	0	0	NA	1296 DEAN STREET	11216		0	0
1019051	2	505000	17847	23970	1620	1620	1017	1639 COLEMAN STREET	11234		18	38
1029659	4	118391	6237	53276	0	41860	5113	50-01 50 AVENUE	11101		0	0

Table 6. Top 10 Fraud Score of Autoencoder Algorithm

RECORD	BBLE	BORO	X	BLOCK	LOT	EASEMENT	OWNER	BLDGCL	TAXCLASS	LTFRONT	LTDEPTH
205992	3035490046	3	Brooklyn	3549	46		YVONNE MORGAN	C3	2A	24	100
409017	4010440001E	4	Queens	1044	1	E	NYC MARINE & AVIATION	Z7	4	6	50
412707	4120130034	4	Queens	12013	34		PARKER, SHAWNESE Y	V0	1B	30	100
476422	4097780134	4	Queens	9778	134			B1	1	46	40
700160	4026460036	4	Queens	2646	36		PIOTR ZUKOWSKI	B1	1	25	119
758435	5023860040	5	Staten Island	2386	40		BENDER, SUSAN	B9	1	31	100
802546	4000341080	4	Queens	34	1080			R4	2	0	0
980961	3009570062	3	Brooklyn	957	62		PRESIDENTIAL OWNERS C	D4	2	124	95
1015400	3012141005	3	Brooklyn	1214	1005			R1	2C	0	0
1019051	3078650023	3	Brooklyn	7865	23		YOUNGSTEIN LARRY	A1	1	30	100

RECORD	STORIES	FULLVAL	AVLAND	AVTOT	EXLAND	EXTOT	EXCD1	STADDR	ZIP	EXMPTCL	BLDFRONT	BLDDEPTH
205992	2	585000	9469	41847	2090	34468	1017	204 TAPSCOTT STREET	11212		24	78
409017	NA	2900	1305	1305	1305	1305	2198	24 AVENUE	NA	X1	0	0
412707	NA	195000	2638	2638	0	0	NA	142 STREET	10467		0	0
476422	2	534000	13539	24675	0	0	NA	158-20 85 AVENUE	11432		27	21
700160	2	478000	13117	23223	1620	1620	1017	56-20 62 AVENUE	11378		20	52
758435	2	410000	14221	23322	0	0	NA	322 NOME AVENUE	10314		22	55
802546	12	163831	6198	73724	2793	70319	5113	5-49 BORDEN AVENUE	11101		0	0
980961	4	1890000	206550	850500	45980	45980	1017	759 PRESIDENT STREET	11215		25	90
1015400	5	133212	3841	19346	0	0	NA	1296 DEAN STREET	11216		0	0
1019051	2	505000	17847	23970	1620	1620	1017	1639 COLEMAN STREET	11234		18	38

CONCLUSION

From the Data Evaluation, we found that of the top 1% of all records, 80% of the records between the two fraud score methods matched in both ranking and fraud score. This shows that both the Heuristic and the Autoencoder Algorithmic approaches performed well in capturing anomalies in the New York Property dataset and were useful in detecting potential fraud. The considerably high overlap percentage of 80% also indicates high accuracy across both algorithms in calculating the fraud scores.

RESULTS

We examined the Top 10 records with the highest fraud scores from each algorithm, listed in Tables 5 and 6, and noticed several anomalies in certain fields, described below:

OWNER

1. We first noticed that the following values in OWNER from both tables were:

   NYC MARINE & AVIATION

   PRESIDENTIAL OWNERS C

   PIOTR ZUKOWSKI

   YOUNGSTEIN LARRY

2. Most of the values in OWNER are not actual name of people but name of companies or government entities.
3. Both fraud methods captured these 3 missing values in OWNER with the following Record IDs:
   476422
   802546
   1015400

   The Heuristic Algorithm captured an additional record with a missing value in OWNER, with Record ID 1029659.

EASEMENT

1. Under both algorithms, 90% of the top 10 records have NO EASEMENT. The single record with Easement of level E was NYC MARINE & AVIATION, which indicates it “has a Land Easement”.

EXMPTCL

1. For both tables, 90% of the top 10 records have no value for EXMPTCL. Again, NYX MARINE & AVIATION is the only property owner with a value of X1 for EXMPTCL.

TAXCLASS

1. In Table 5, Top 10 Fraud Scores of the Heuristic Algorithm, two records belong to tax class 2C. This is unusual because within the New York Property Dataset, this is an uncommon tax class.

BORO

1. Most of the highest fraud scores come from BORO 3 and 4, which indicate the code for Brooklyn and Queens respectively.
2. OWNER PRIME EAST REALTY LLC is the only value that comes from BORO 1 from the top 10 Fraud Scores, which indicates the code for Manhattan.
3. From table 6, OWNER BENDER, SUSAN is the only value that comes from BORO 5 from the top 10 Fraud Scores, which indicates the code for Staten Island

FULLVAL, AVLAND, AVTOT

1. OWNER NYC MARINE & AVIATION has unusually small values for FULLVAL, AVLAND, and AVTOT for a business entity.
2. OWNER PRIME EAST REALTY LLC has the highest value for FULLVAL and OWNER PRESIDENTIAL OWNERS C has the highest values for AVLAND and AVTOT.

EXLAND, EXTOT

1. In Table 5, half of the top 10 records have a value of 0 for EXLAND and EXTOT, indicating these records have no exempt land and a land value of 0 for exempt property. In Table 4, only 4 of the top 10 records have a value of 0 for EXLAND and EXTOT.

STORIES

1. Most of the missing values for STORIES come from missing values of OWNER.
2. Among the top 10 records in both tables, one property has 12 STORIES, while all other properties have 5 or less STORIES.

LTFRONT, LTDEPTH, BLDFRONT, BLDDEPTH

1. Most of the values of 0 for LTFRONT and LTDEPTH only come from missing values of OWNER.
2. OWNER PRESIDENTIAL OWNERS C contains values of 0 for BLDFRONT and BLDDEPTH

ZIP

1. The only missing value for ZIP comes from OWNER PRESIDENTIAL OWNERS C.
2. The majority of the first two digit values for ZIP start with 11.

STADDR

1. OWNER PRESIDENTIAL OWNERS C resides in STADDR President Street. According to Google Maps, President Street does exist in Brooklyn but the OWNER name is an usually perfect match with this street name.

FRAUD SCENARIOS

From the results listed above, we can assume the following possible fraud scenarios:

1. Mortgage Fraud: Incorrect data entry by individuals (OWNER) in certain fields such as FULLVAL and AVLAND in order to get qualified for loans.
2. Tax Fraud: Individual or business entity (OWNER) willfully and intentionally falsified information on a tax return (TAXCLASS) in order to limit the amount of tax liability.

POTENTIAL IMPROVEMENT

Performing fraud analysis on the New York Property dataset consisted of multiple phases, including understanding, cleaning, modeling, and evaluating the New York Property dataset.

Throughout the overall process, the following areas could have significantly improved our fraud analysis on the New York Property dataset:

1. **Better Understanding of the Dataset:**
   Clearer definition of each variable could have led us to a more efficient way of handling missing values.
2. **Access to More Data:**
   Data is critical for modeling and building good machine learning algorithms. The more data we have, the better our model is when it comes to fraud detection.
3. **Performing More Unsupervised Learning Algorithm Methods:**
   Using additional approaches to calculate the fraud score would have provided further insights on anomalies in the dataset. For further investigation we could have used methods such as:
   a. MeanShift - A non-parametric clustering algorithm that discovers groups in datasets of smooth density by finding the maximum of a density function
   b. Gaussian Mixture Model (GMM) - A clustering algorithm that computes clusters by fitting a given number of Gaussians to the dataset and iteratively estimating their parameters

APPENDIX

DATA QUALITY REPORT

DATA BASIC INFORMATION

DATA SOURCE

The dataset comes from Department of Finance(DOF) of New York City. You can download it from NYC OpenData. Here is the link: https://data.cityofnewyork.us/Housing-Development/Property-Valuation-and-Assessment-Data/rgy2-tti8

DATA TIME

The dataset was first created at September 2, 2011. It was most recently updated at December 22, 2017. The data quality report is based on its version updated on Nov,2011.

DATA SIZE

The dataset has 1048575 observations with 30 variables.

GLANCE OF THE DATA

	0	1	2
RECORD	1	2	3
BBLE	3046020035	5046820019	3074790028
BLOCK	4602	4682	7479
LOT	35	19	28
EASEMENT	NaN	NaN	NaN
OWNER	DESMOND CAMPBELL	CINISOMO MARIO	GANGICHIODO DONALD
BLDGCL	B1	A5	V0
TAXCLASS	1	1	1B
LTFRONT	18	25	16
LTDEPTH	100	100	19
STORIES	2	3	NaN
FULLVAL	407000	415000	128000
AVLAND	12337	13301	81
AVTOT	19537	21312	81
EXLAND	1620	1620	0
EXTOT	1620	1620	0
EXCD1	1017	1017	NaN
STADDR	140 EAST 49 STREET	537 AMHERST AVENUE	COYLE STREET
ZIP	11203	10306	NaN
EXMPTCL	X7	NaN	NaN
BLDFRONT	18	14	0
BLDDEPTH	36	51	0
AVLAND2	NaN	NaN	NaN
AVTOT2	NaN	NaN	NaN
EXLAND2	NaN	NaN	NaN
EXTOT2	NaN	NaN	NaN
EXCD2	NaN	NaN	NaN
PERIOD	FINAL	FINAL	FINAL
YEAR	2010/11	2010/11	2010/11
VALTYPE	AC-TR	AC-TR	AC-TR

	3	4	5
RECORD	4	5	6
BBLE	4027980132	1006950027E	4031810007
BLOCK	2798	695	3181
LOT	132	27	7
EASEMENT	NaN	E	NaN
OWNER	DCAS	CONRAIL	BERGERSON ERIC W
BLDGCL	V0	U6	A5
TAXCLASS	1B	3	1
LTFRONT	21	0	20
LTDEPTH	75	0	100
STORIES	NaN	NaN	2
FULLVAL	112613	0	582000
AVLAND	1940	0	17802
AVTOT	1940	0	29859
EXLAND	0	0	0
EXTOT	0	0	0
EXCD1	NaN	NaN	NaN
STADDR	MAZEAU STREET	WEST 23 STREET	90-07 68 AVENUE
ZIP	NaN	NaN	11375
EXMPTCL	NaN	NaN	NaN
BLDFRONT	0	0	20
BLDDEPTH	0	0	37
AVLAND2	NaN	NaN	NaN
AVTOT2	NaN	NaN	NaN
EXLAND2	NaN	NaN	NaN
EXTOT2	NaN	NaN	NaN
EXCD2	NaN	NaN	NaN
PERIOD	FINAL	FINAL	FINAL
YEAR	2010/11	2010/11	2010/11
VALTYPE	AC-TR	AC-TR	AC-TR

	6	7	8
RECORD	7	8	9
BBLE	4051861001	3082020064	4052570008
BLOCK	5186	8202	5257
LOT	1001	64	8
EASEMENT	NaN	NaN	NaN
OWNER	GOLDEN HUANG LLC	SPICER, CLINTON	SILVIA SIPAVICIUS
BLDGCL	R5	B1	A1
TAXCLASS	4	1	1
LTFRONT	0	24	40
LTDEPTH	0	100	96
STORIES	6	2	2
FULLVAL	539000	416000	660000
AVLAND	30960	13966	14418
AVTOT	242550	22345	38064
EXLAND	0	0	0
EXTOT	0	0	0
EXCD1	NaN	NaN	NaN
STADDR	43-55 KISSENA BOULEVARD	1200 EAST 95 STREET	172-16 33 AVENUE
ZIP	11355	11236	11358
EXMPTCL	NaN	NaN	NaN
BLDFRONT	0	20	21
BLDDEPTH	0	44	49
AVLAND2	30960	NaN	NaN
AVTOT2	268740	NaN	NaN
EXLAND2	NaN	NaN	NaN
EXTOT2	NaN	NaN	NaN
EXCD2	NaN	NaN	NaN
PERIOD	FINAL	FINAL	FINAL
YEAR	2010/11	2010/11	2010/11
VALTYPE	AC-TR	AC-TR	AC-TR

VARIABLE INFORMATION

	Variable Type	Percent Populated (%)
RECORD	numeric	100
BBLE	numeric	100
BLOCK	numeric	100
LOT	numeric	100
EASEMENT	categorical	39
OWNER	string	97
BLDGCL	categorical	100
TAXCLASS	categorical	100
LTFRONT	numeric	100
LTDEPTH	numeric	100
STORIES	numeric	95
FULLVAL	numeric	100
AVLAND	numeric	100
AVTOT	numeric	100
EXLAND	numeric	100

EXTOT	numeric	100
EXCD1	numeric	59
STADDR	string	99.9
ZIP	numeric	97.5
EXMPTCL	categorical	1.4
BLDFRONT	numeric	100
BLDDEPTH	numeric	100
AVLAND2	numeric	26.8
AVTOT2	numeric	26.8
EXLAND2	numeric	8.3
EXTOT2	numeric	12.4
EXCD2	numeric	8.7
PERIOD	categorical	100
YEAR	date	100
VALTYPE	categorical	100

DESCRIPTIVE STATISTICS FOR NUMERIC VARIABLES

	RECORD	BLOCK	LOT	LTFRONT	LTDEPTH
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06
mean	5.242880e+05	4.708867e+03	3.700924e+02	3.617425e+01	8.827643e+01
std	3.026977e+05	3.699547e+03	8.605382e+02	7.373356e+01	7.547885e+01
min	1.000000e+00	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
25%	2.621445e+05	1.534000e+03	2.300000e+01	1.900000e+01	8.000000e+01
50%	5.242880e+05	3.944000e+03	4.900000e+01	2.500000e+01	1.000000e+02
75%	7.864315e+05	6.797000e+03	1.460000e+02	4.000000e+01	1.000000e+02
max	1.048575e+06	1.635000e+04	9.978000e+03	9.999000e+03	9.999000e+03

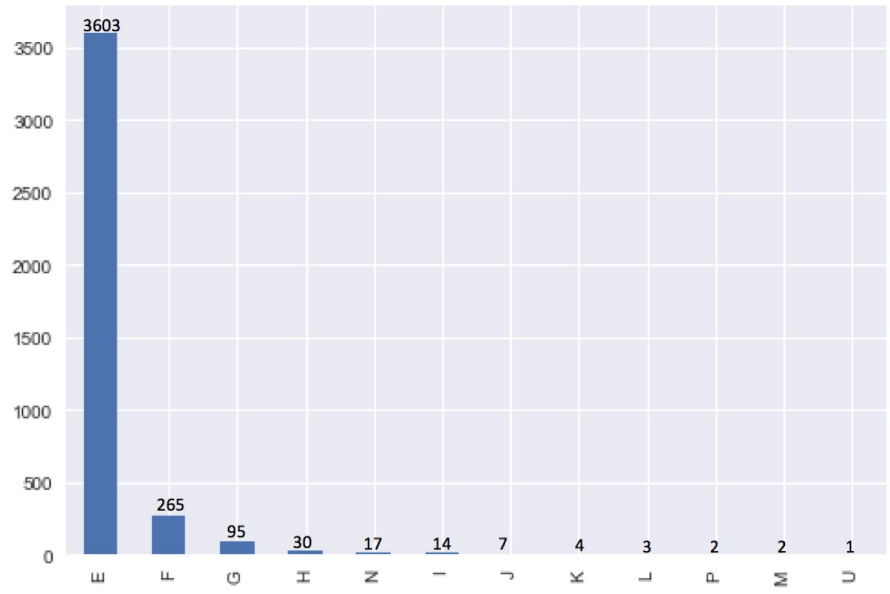
	STORIES	FULLVAL	AVLAND	AVTOT	EXLAND
count	996433.000000	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06
mean	5.063363	8.804877e+05	8.599503e+04	2.307582e+05	3.681179e+04
std	8.431372	1.170293e+07	4.100755e+06	6.951206e+06	4.024330e+06
min	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.000000	3.030000e+05	9.160000e+03	1.838500e+04	0.000000e+00
50%	2.000000	4.460000e+05	1.364600e+04	2.533900e+04	1.620000e+03
75%	3.000000	6.190000e+05	1.970600e+04	4.609500e+04	1.620000e+03
max	119.000000	6.150000e+09	2.668500e+09	4.668309e+09	2.668500e+09

	EXTOT	EXCD1	ZIP	BLDFRONT	BLDDEPTH
count	1.048575e+06	622642.000000	1.022219e+06	1.048575e+06	1.048575e+06
mean	9.254381e+04	1604.500100	1.093532e+04	2.301872e+01	4.007421e+01
std	6.578281e+06	1388.131676	5.265759e+02	3.578847e+01	4.303640e+01
min	0.000000e+00	1010.000000	1.000100e+04	0.000000e+00	0.000000e+00
25%	0.000000e+00	1017.000000	1.045300e+04	1.500000e+01	2.600000e+01
50%	1.620000e+03	1017.000000	1.121500e+04	2.000000e+01	3.900000e+01
75%	2.090000e+03	1017.000000	1.136400e+04	2.400000e+01	5.100000e+01
max	4.668309e+09	7170.000000	3.380300e+04	7.575000e+03	9.393000e+03

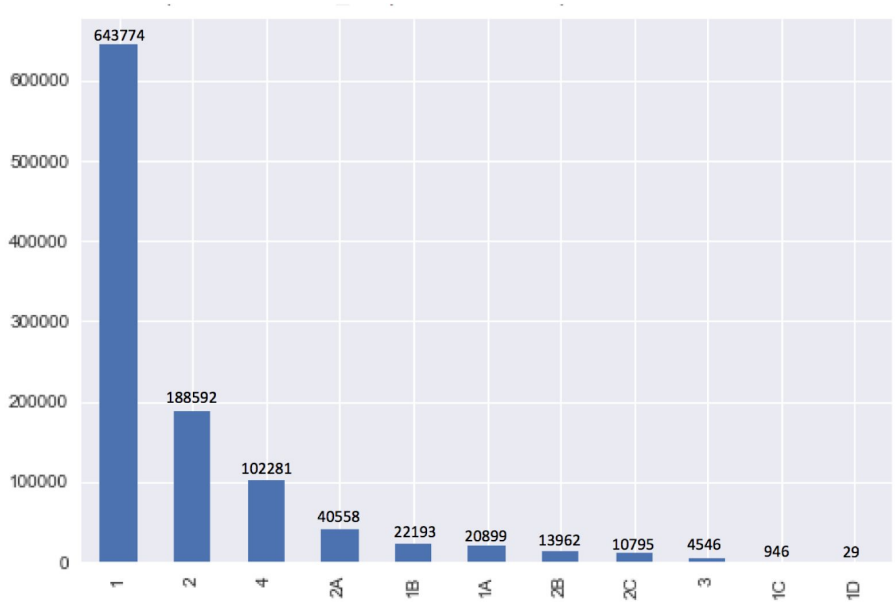
	AVLAND2	AVTOT2	EXLAND2	EXTOT2	EXCD2
count	2.809660e+05	2.809720e+05	8.667500e+04	1.299330e+05	90941.000000
mean	2.463655e+05	7.160787e+05	3.518022e+05	6.581148e+05	1371.659098
std	6.199390e+06	1.169017e+07	1.085248e+07	1.612981e+07	1105.489791
min	3.000000e+00	3.000000e+00	1.000000e+00	7.000000e+00	1011.000000
25%	5.705000e+03	3.401350e+04	2.090000e+03	2.889000e+03	1017.000000
50%	2.005900e+04	8.001000e+04	3.053000e+03	3.711600e+04	1017.000000
75%	6.233875e+04	2.407920e+05	3.141900e+04	1.066290e+05	1017.000000
max	2.371005e+09	4.501180e+09	2.371005e+09	4.501180e+09	7160.000000

DISTRIBUTIONS OF CATEGORICAL VARIABLES

EASEMENT



TAXCLASS

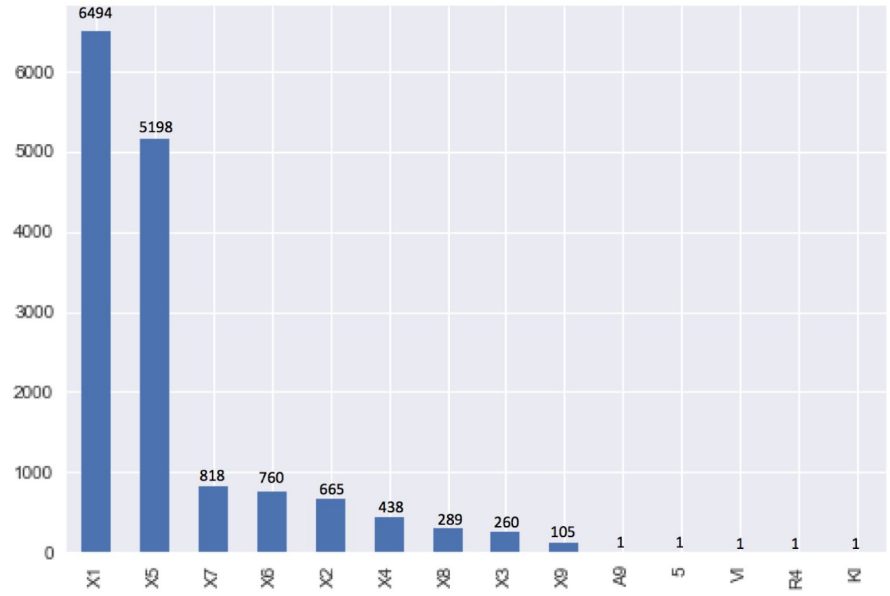


BLDGCL

There are 200 kinds of BLDGCL. The following table shows the top 10.

	BLDGCL	total_count
1	R4	139879
2	A1	119340
3	A5	92896
4	B1	84054
5	B2	73156
6	C0	73077
7	B3	59091
8	A2	49085
9	A9	25931
10	B9	25235

EXMPTCL

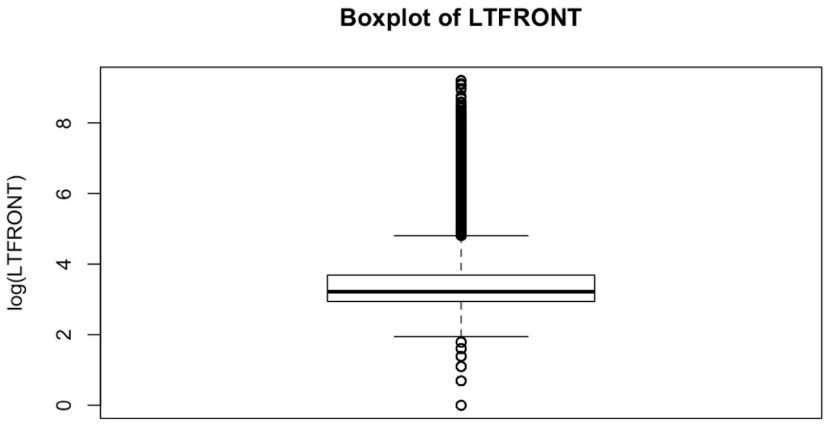


PERIOD & VALTYPE

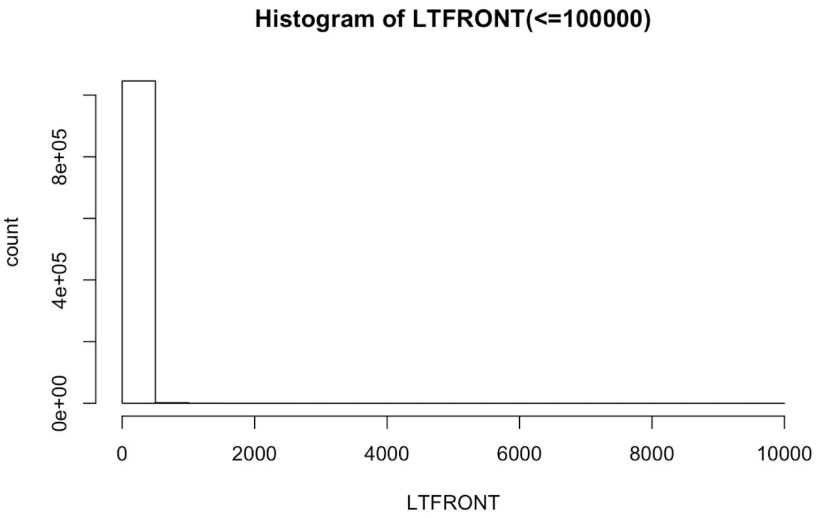
Although these two are also categorical variables, they only have one kind of value, which are not very useful for the modeling, so this report is not going to show the distribution of these two variables.

DISTRIBUTIONS OF NUMERIC VARIABLES

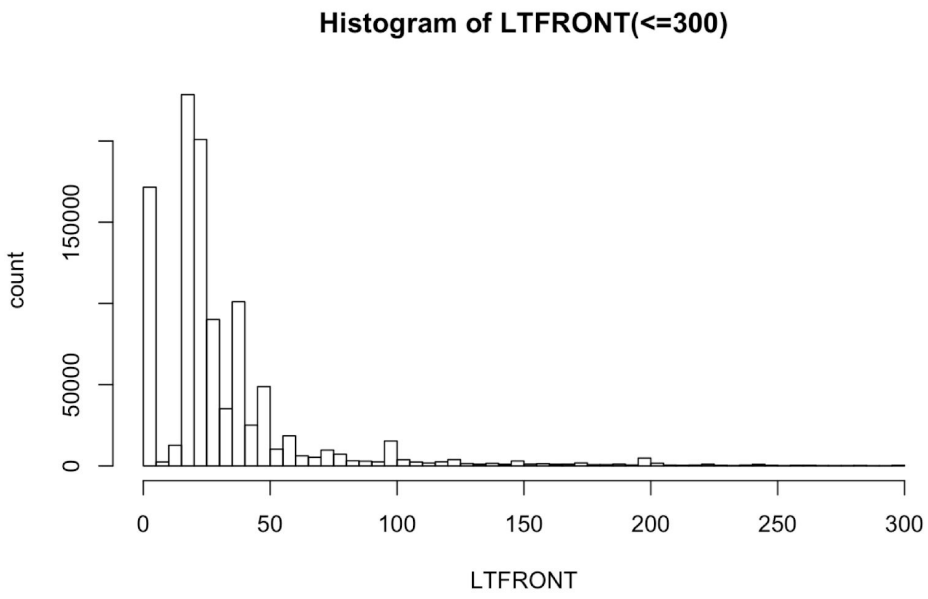
LTFRONT



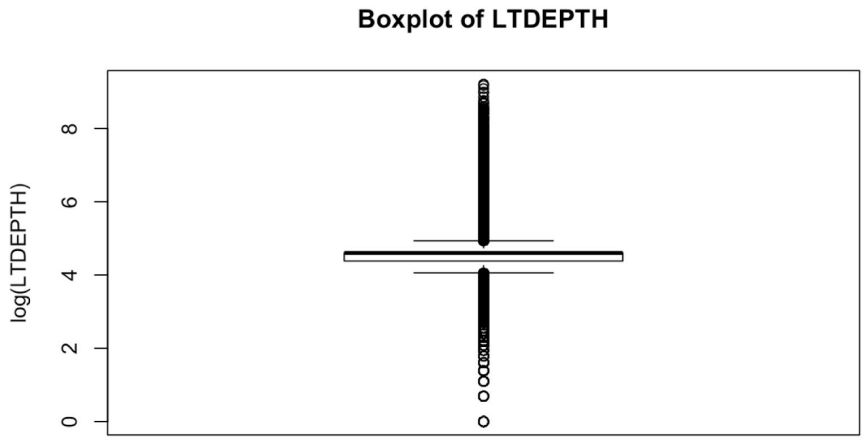
From the boxplot, we observe that 75% of the values of LTFRONT lies between 100 and 100000. So we choose those properties that have LTFRONT less than 100000 to make a histogram to see a detailed distribution.



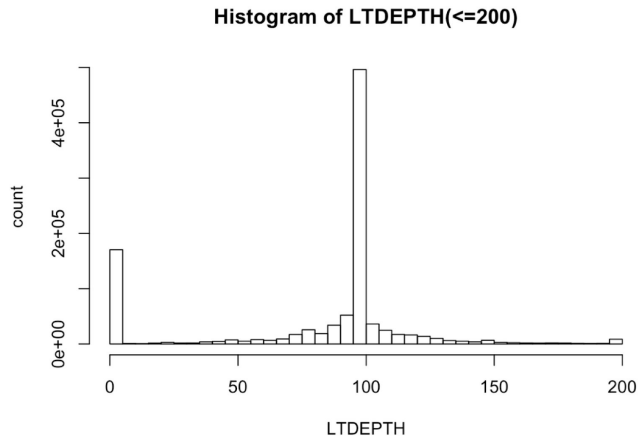
It looks like we still need to narrow the range of LTFRONT.



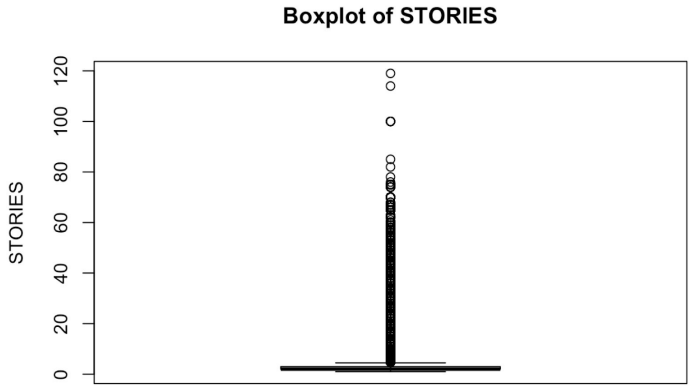
LTDEPTH



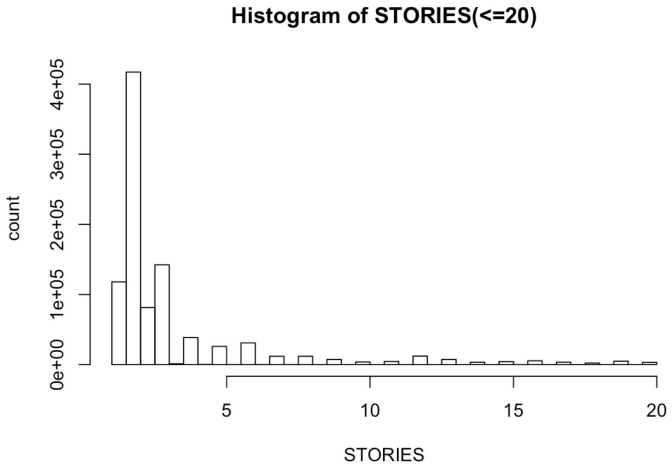
From the boxplot, we can observe that the distribution of LTDEPTH is more centralized than LTFRONT's. After narrowing down the range, we get the following histogram.



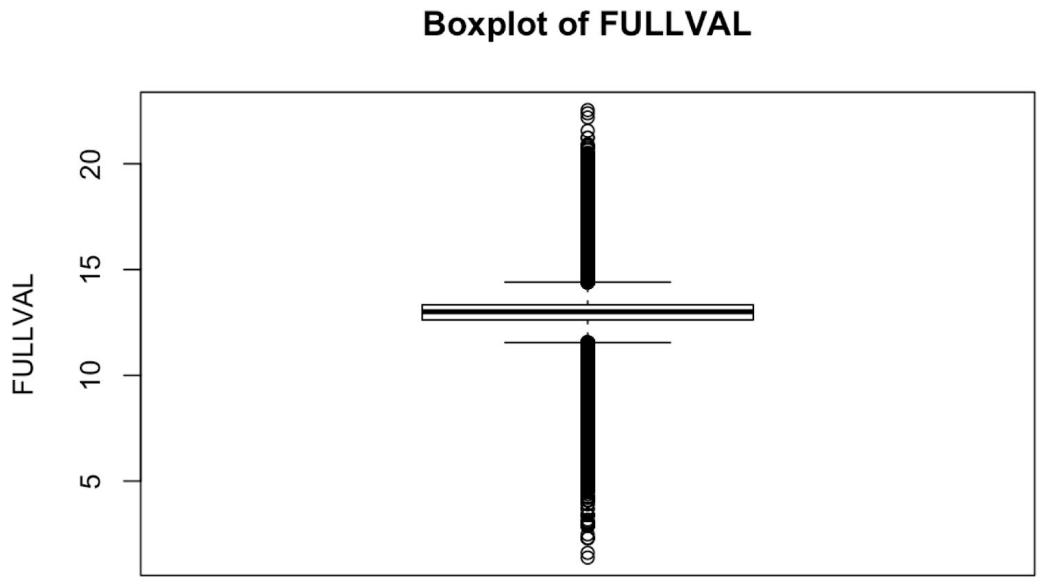
STORIES

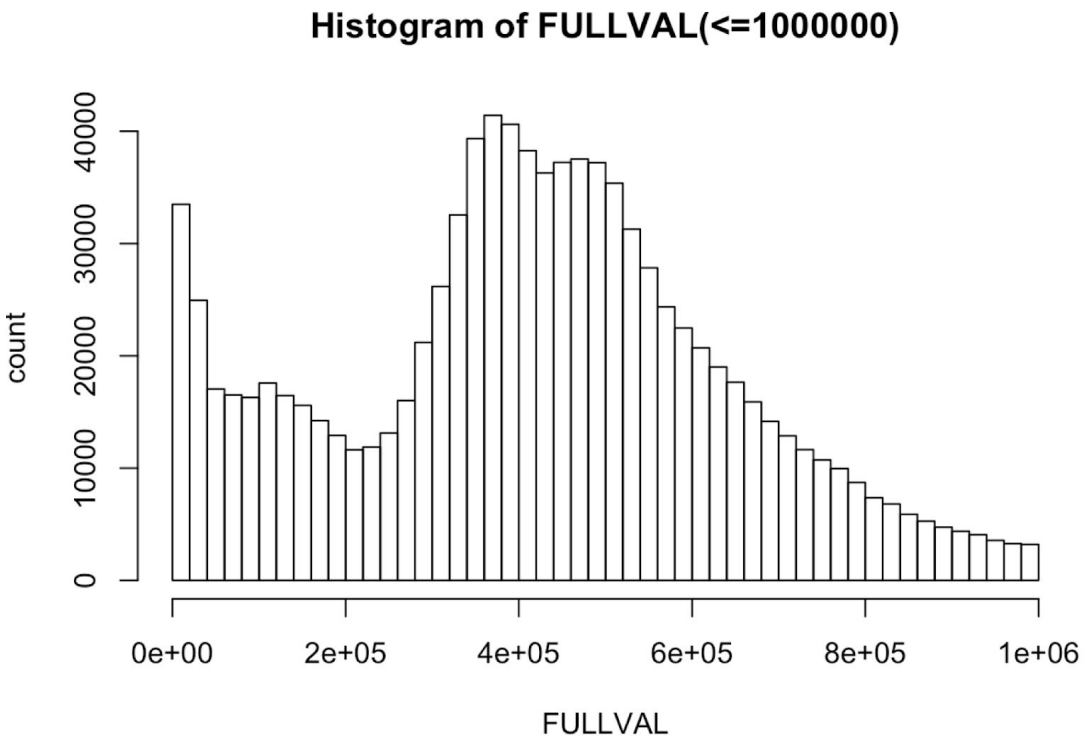


From the boxplot, we can observe that most of properties in NY City are not very high, maybe because most of properties is house or apartment.

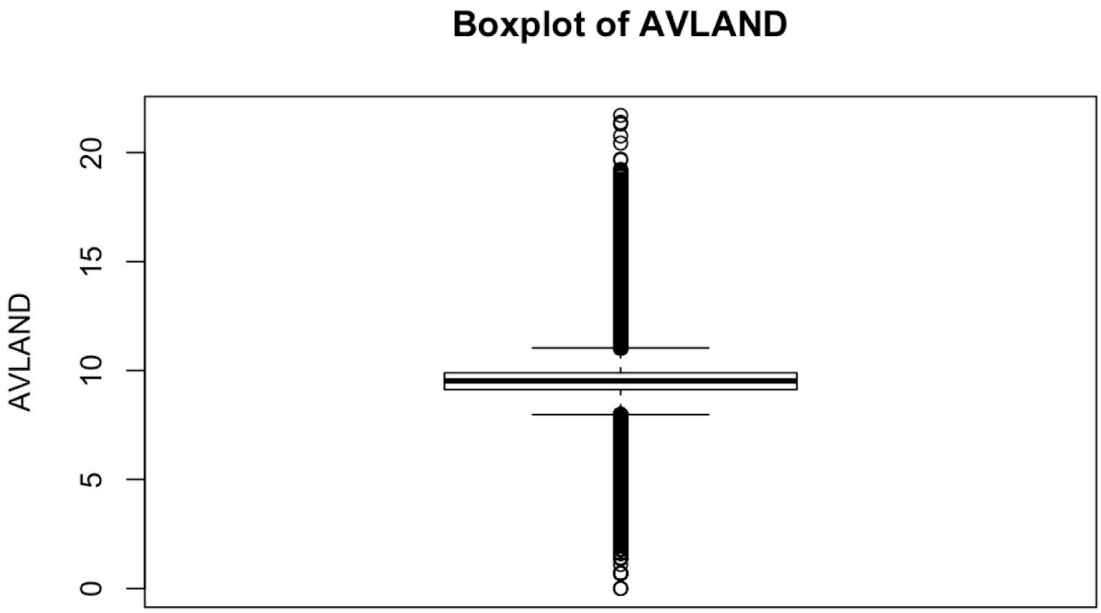


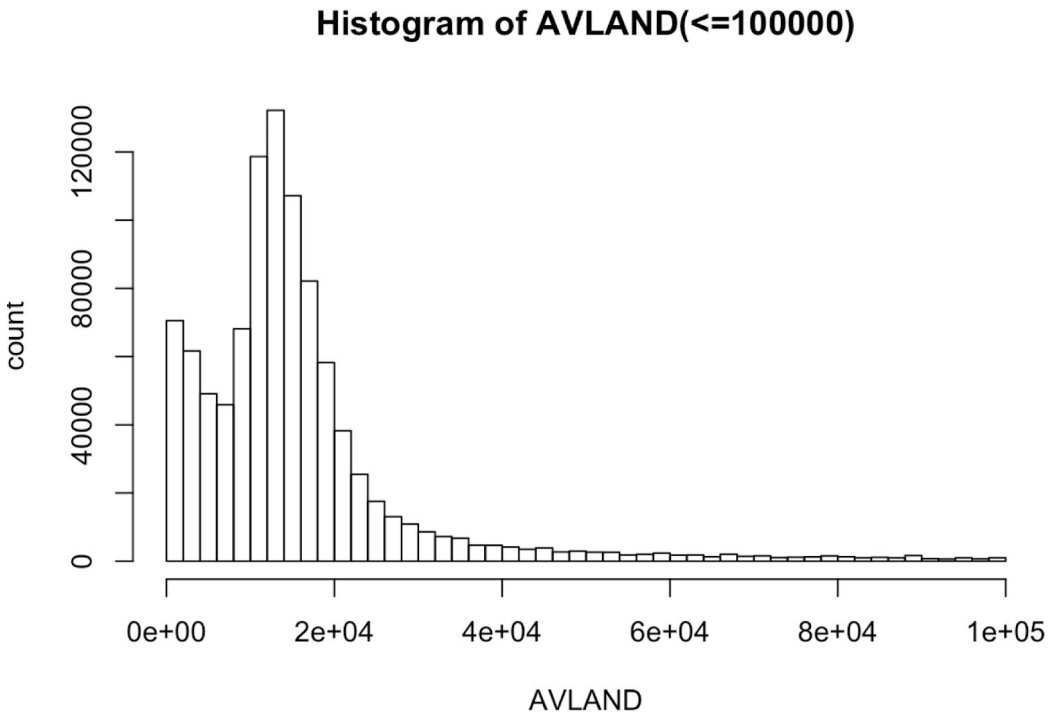
FULLVAL


Boxplot of FULLVAL
FULLVAL
5
10
15
20

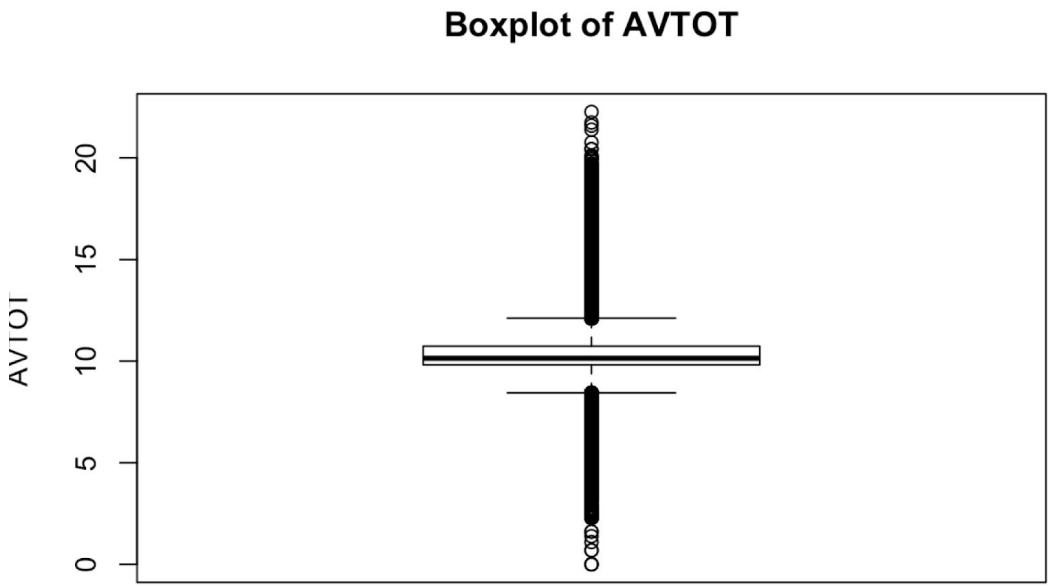

Histogram of FULLVAL(<=1000000)
count
0
10000
20000
30000
40000
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
FULLVAL

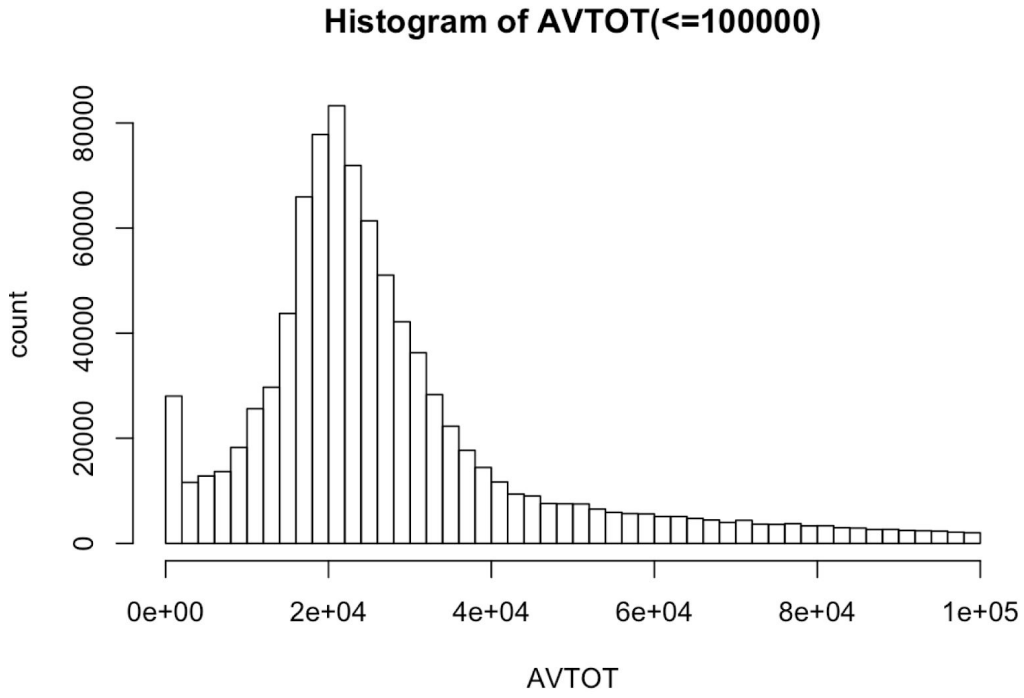
AVLAND


Boxplot of AVLAND
AVLAND
0
5
10
15
20

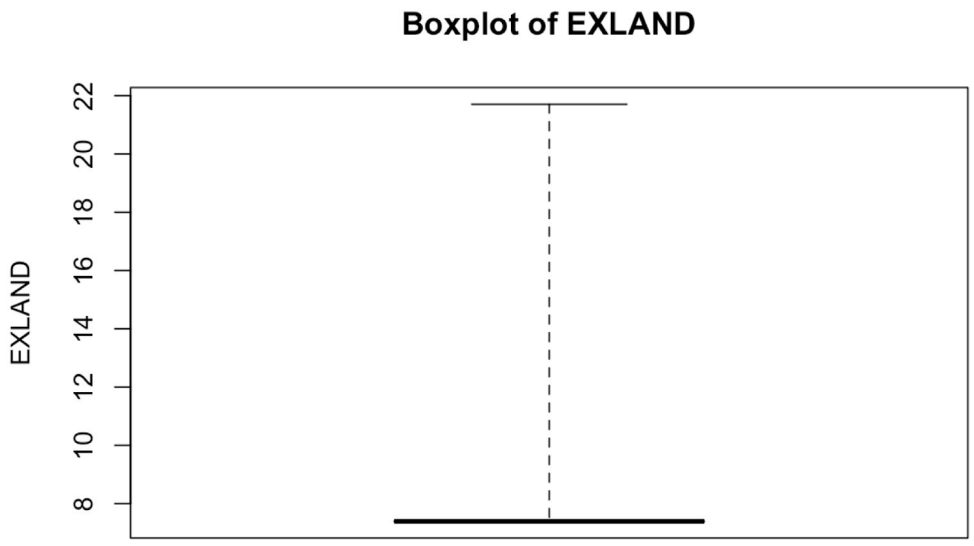

Histogram of AVLAND(<=100000)
count
0
40000
80000
120000
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
AVLAND

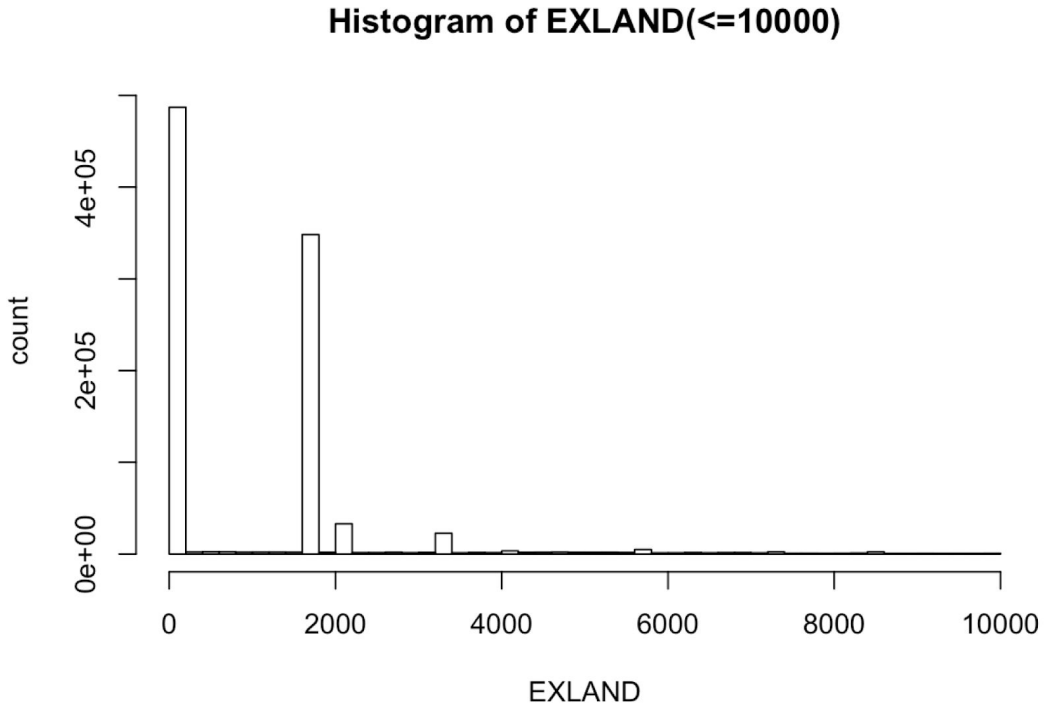
AVTOT



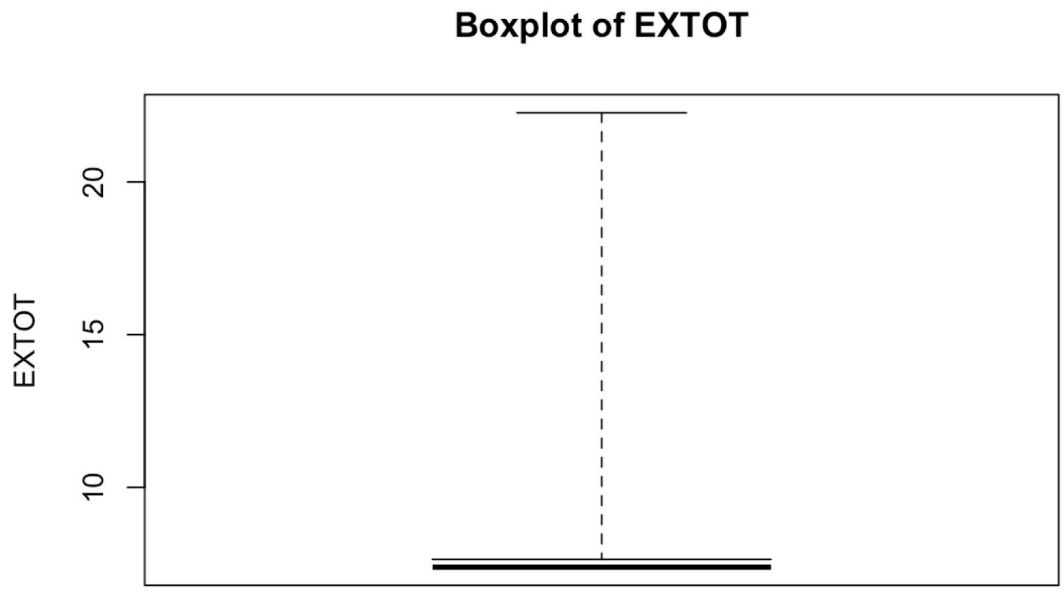


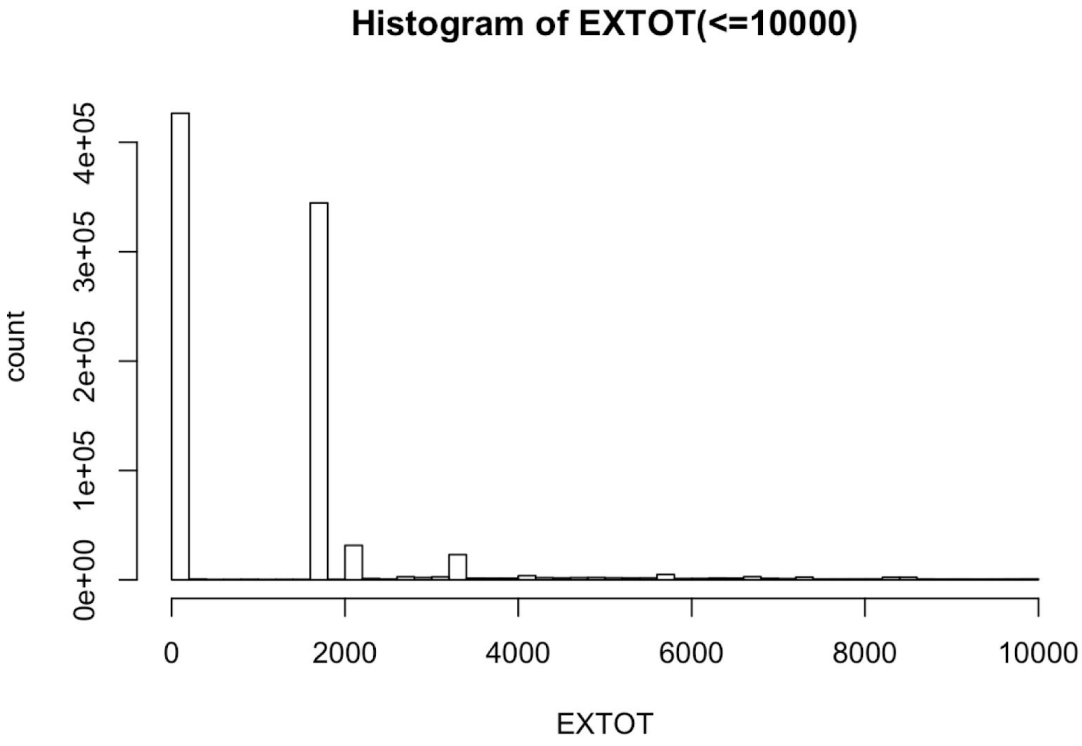
EXLAND


Boxplot of EXLAND
EXLAND
8
10
12
14
16
18
20
22

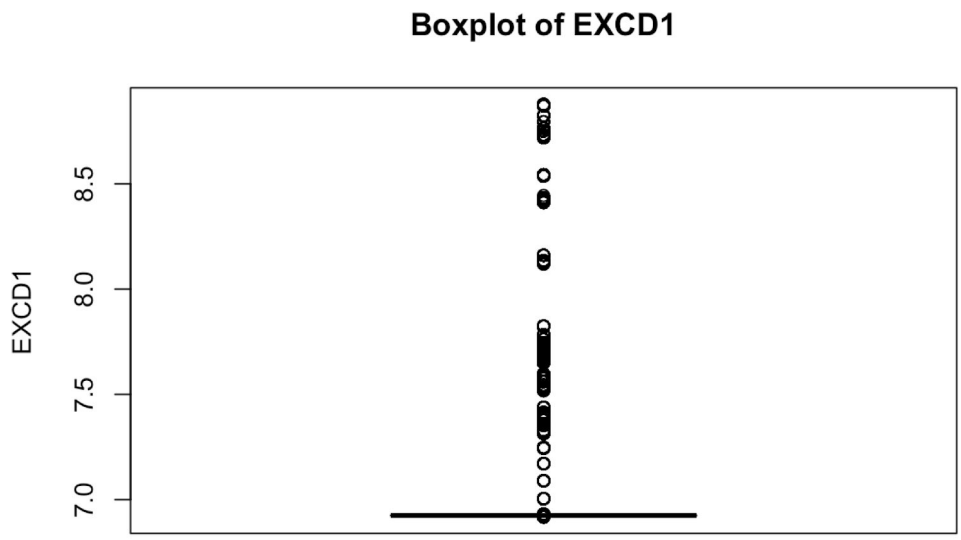

Histogram of EXLAND(<=10000)
count
0e+00
2e+05
4e+05
0
2000
4000
6000
8000
10000
EXLAND

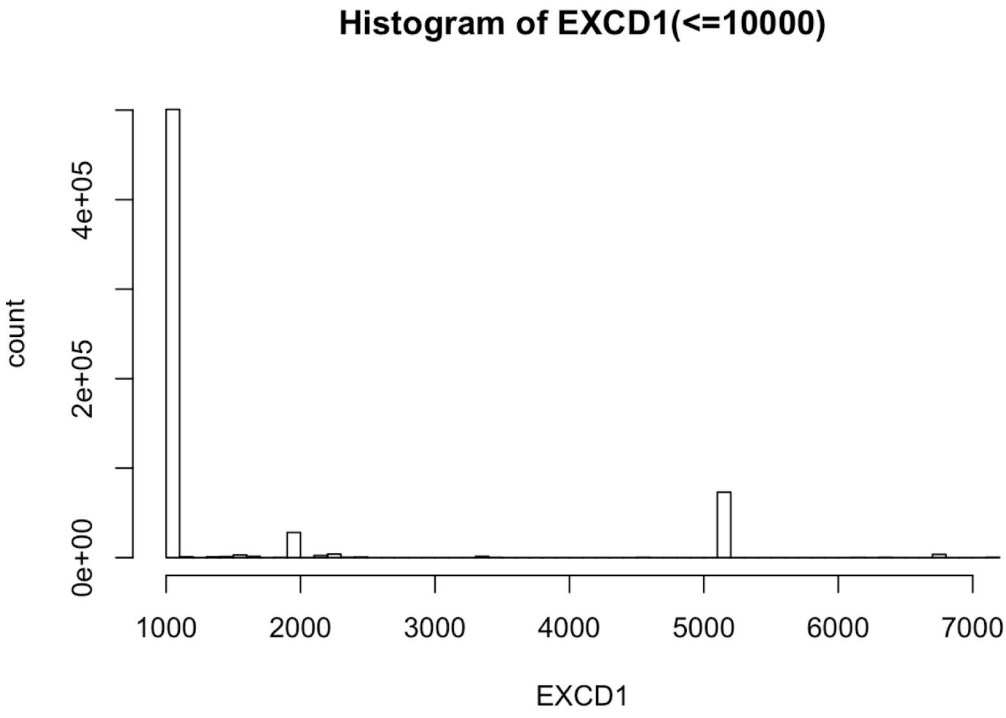
EXTOT


Boxplot of EXTOT
EXTOT
10
15
20

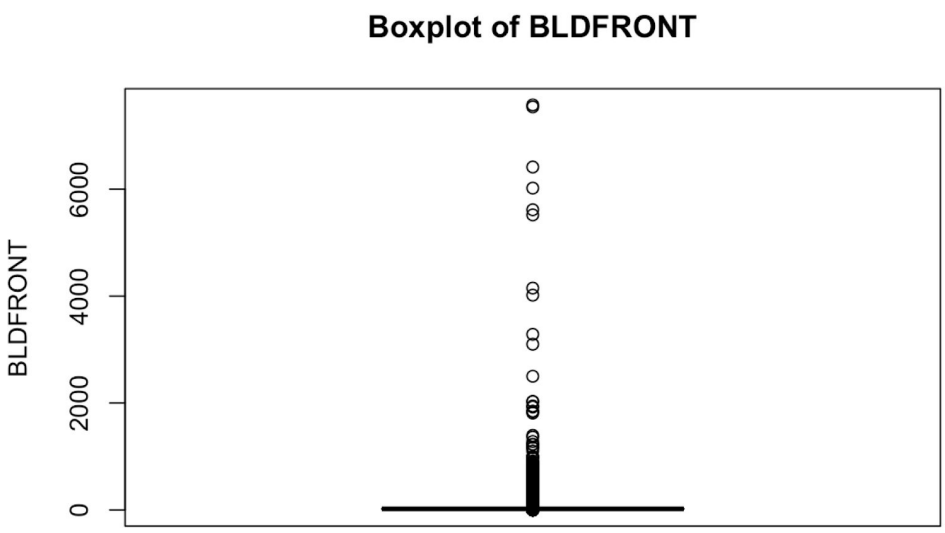

Histogram of EXTOT(<=10000)
count
0e+00
1e+05
2e+05
3e+05
4e+05
0
2000
4000
6000
8000
10000
EXTOT

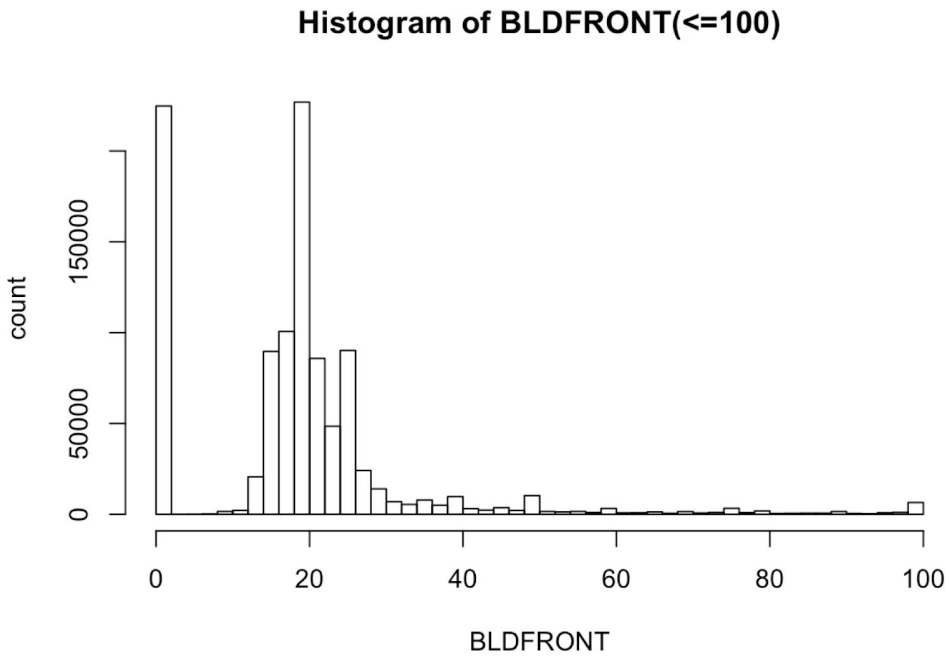
EXCD1


Boxplot of EXCD1
EXCD1
7.0
7.5
8.0
8.5

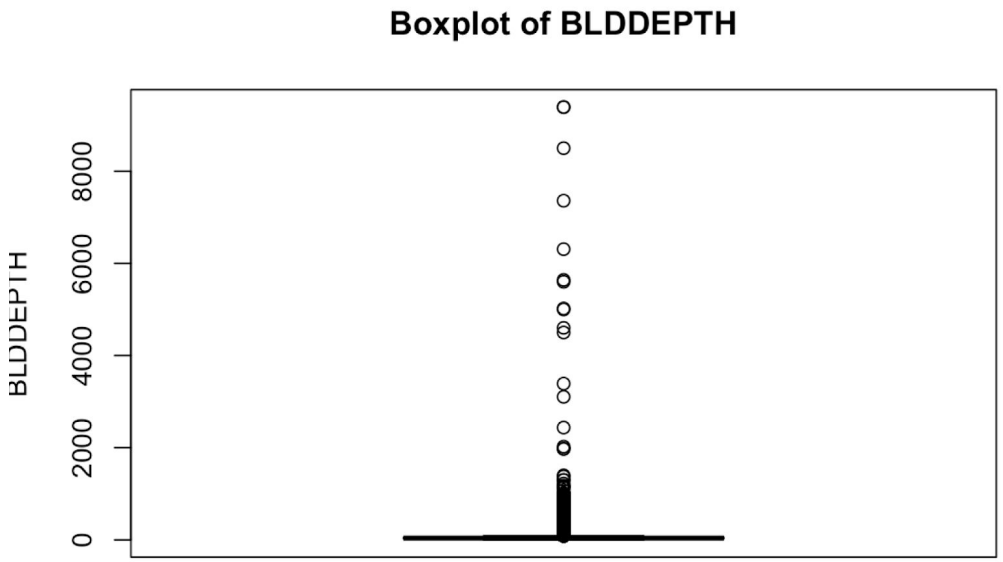

Histogram of EXCD1(<=10000)
count
0e+00
2e+05
4e+05
1000
2000
3000
4000
5000
6000
7000
EXCD1

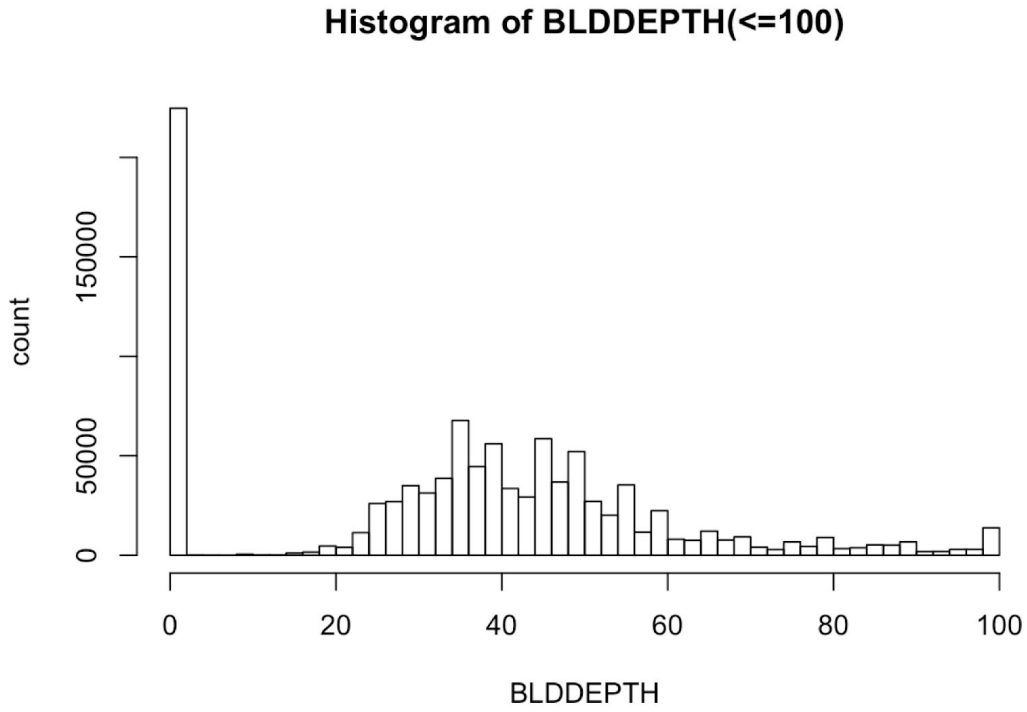
BLDFRONT


Boxplot of BLDFRONT
BLDFRONT
0
2000
4000
6000

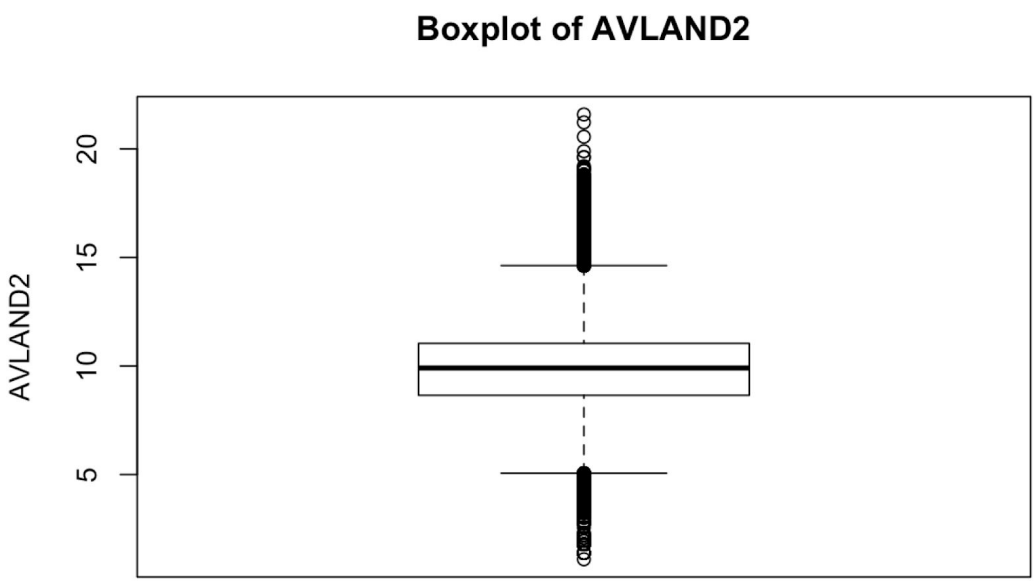

Histogram of BLDFRONT(<=100)
count
0
50000
150000
0
20
40
60
80
100
BLDFRONT

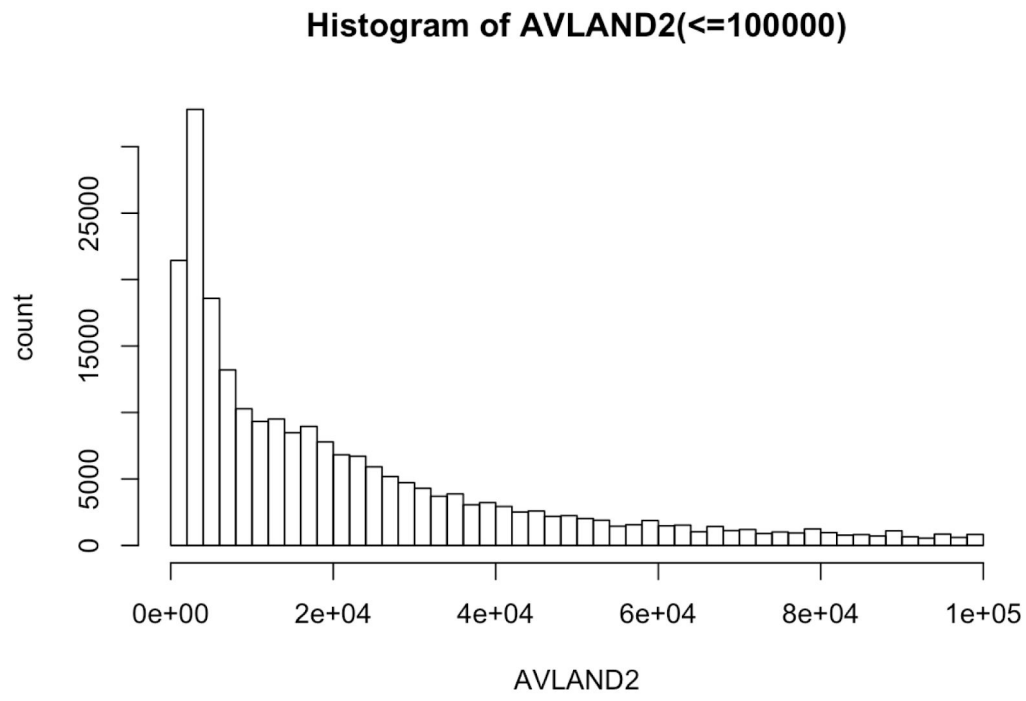
BLDDEPTH


Boxplot of BLDDEPTH
BLDDEPTH
0
2000
4000
6000
8000

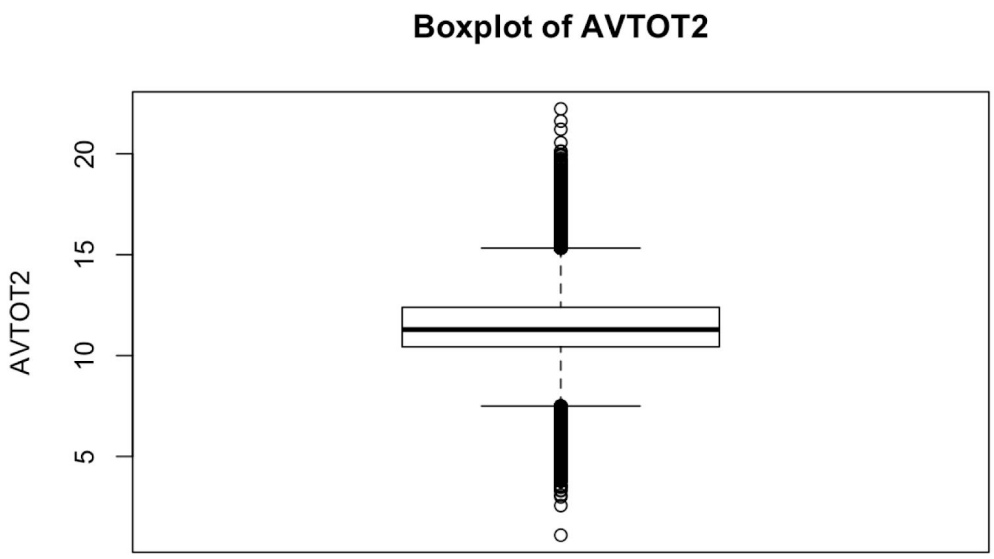

Histogram of BLDDEPTH(<=100)
count
0
50000
150000
0
20
40
60
80
100
BLDDEPTH

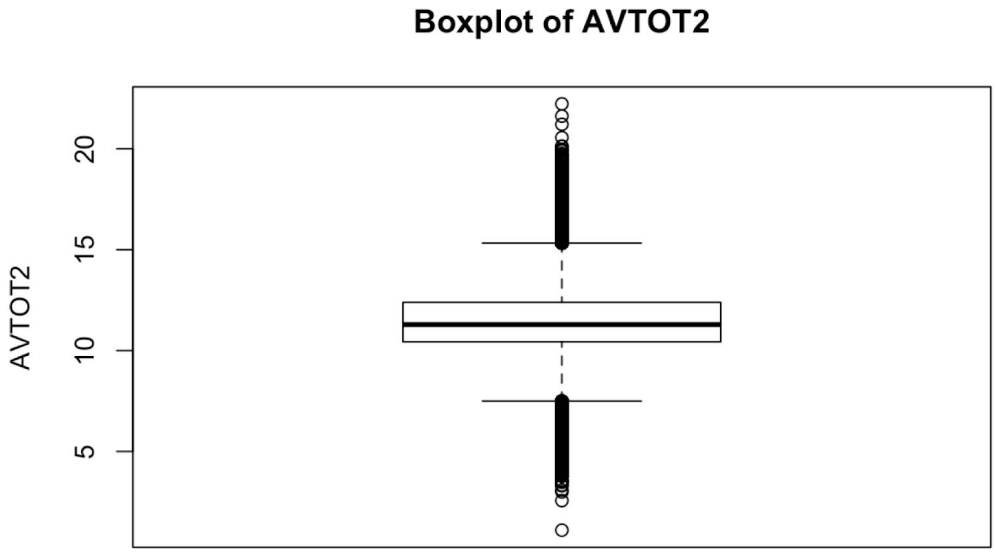
AVLAND2


Boxplot of AVLAND2
AVLAND2
5
10
15
20

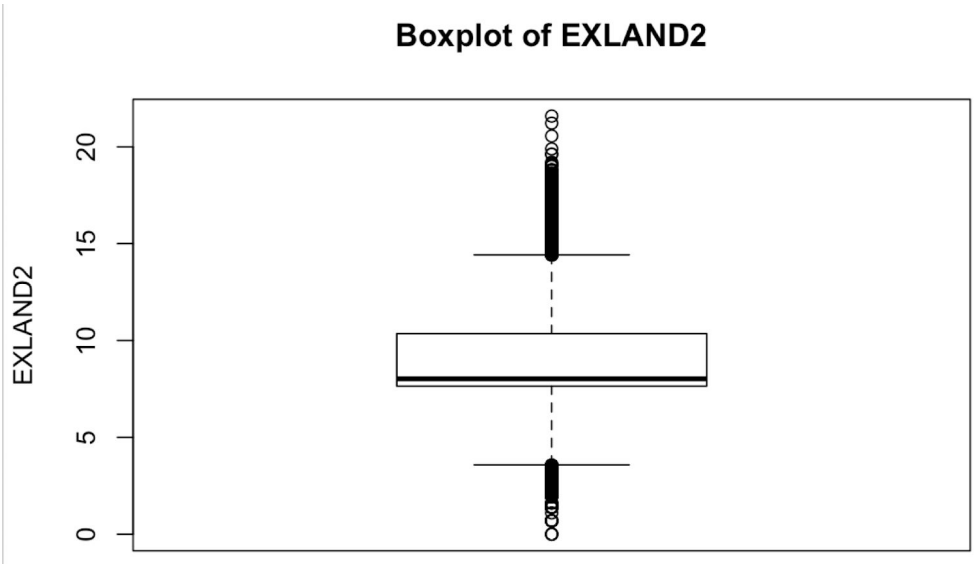

Histogram of AVLAND2(<=100000)
count
0
5000
15000
25000
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
AVLAND2

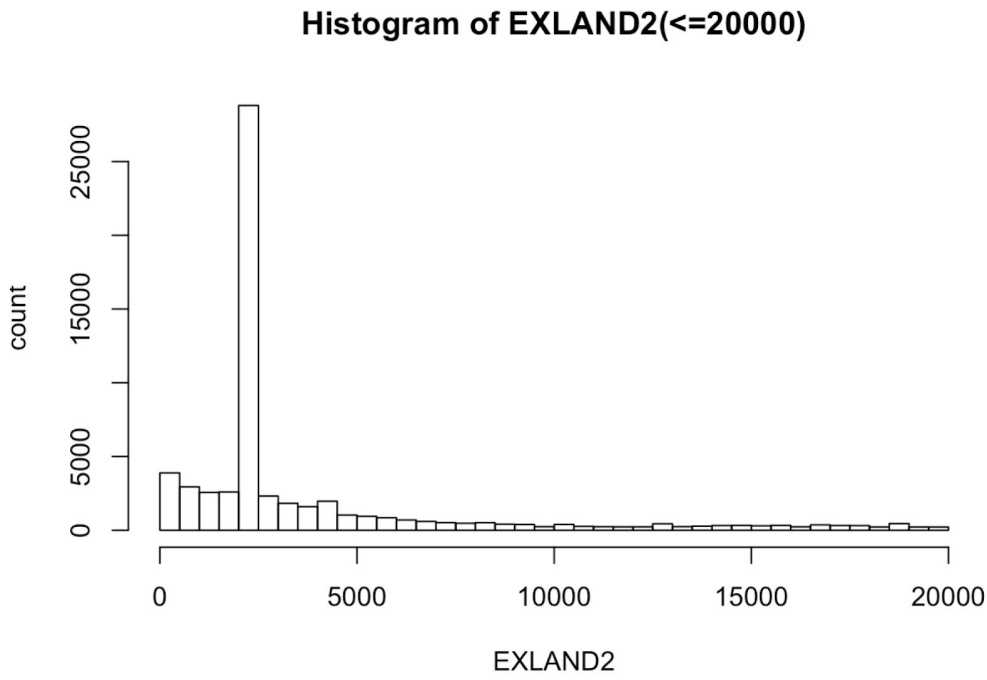
AVTOT2


Boxplot of AVTOT2
AVTOT2
5
10
15
20

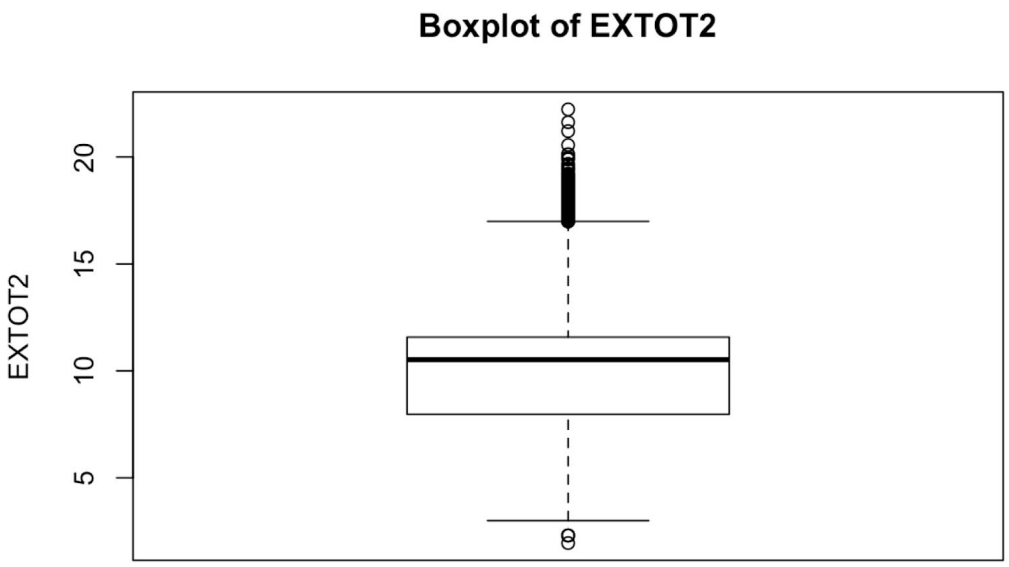

Boxplot of AVTOT2
AVTOT2
5
10
15
20

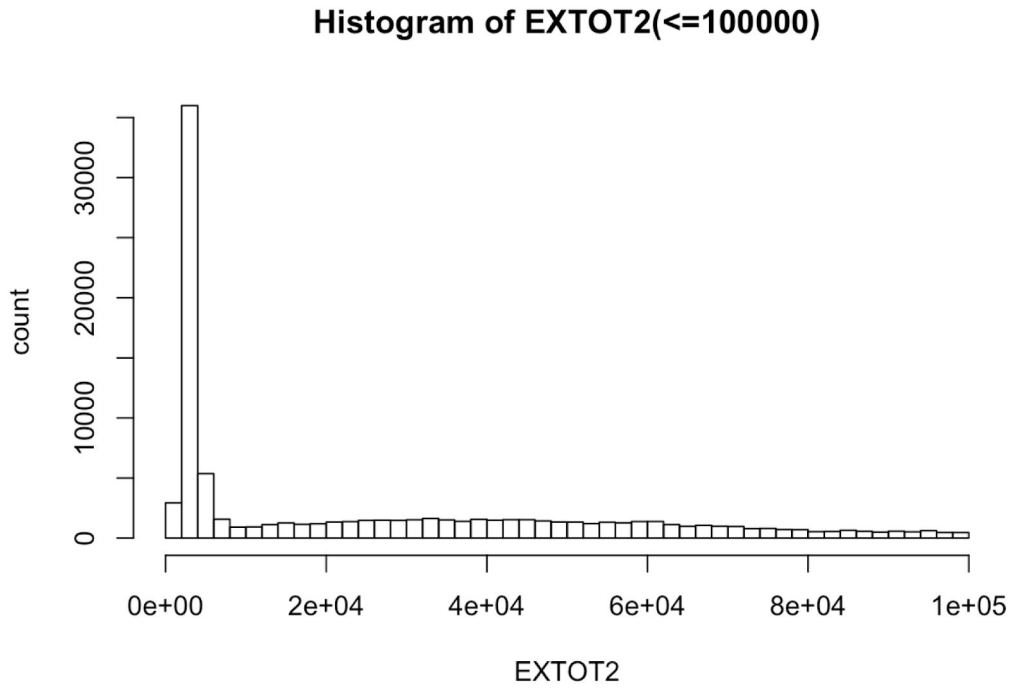
EXLAND2


Boxplot of EXLAND2
EXLAND2
0
5
10
15
20

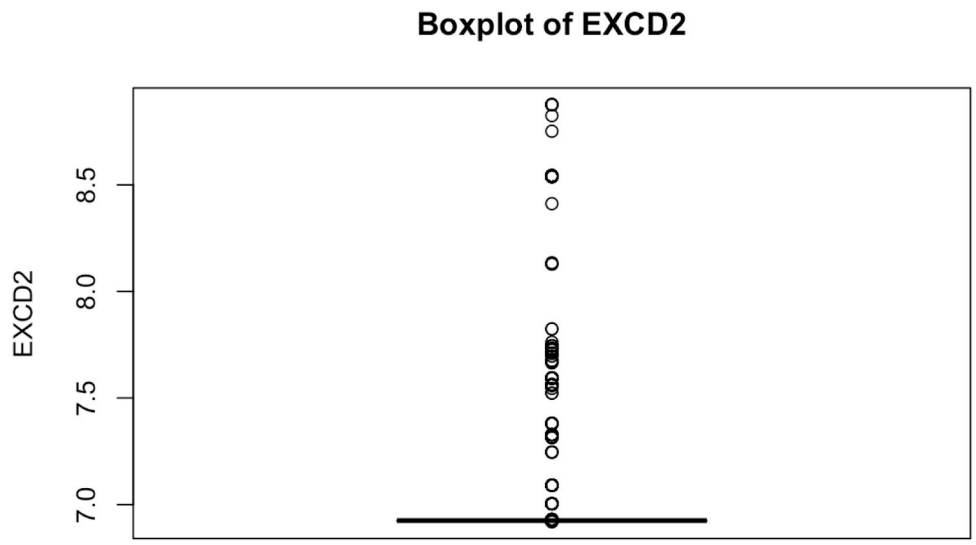

Histogram of EXLAND2(<=20000)
count
0
5000
15000
25000
0
5000
10000
15000
20000
EXLAND2

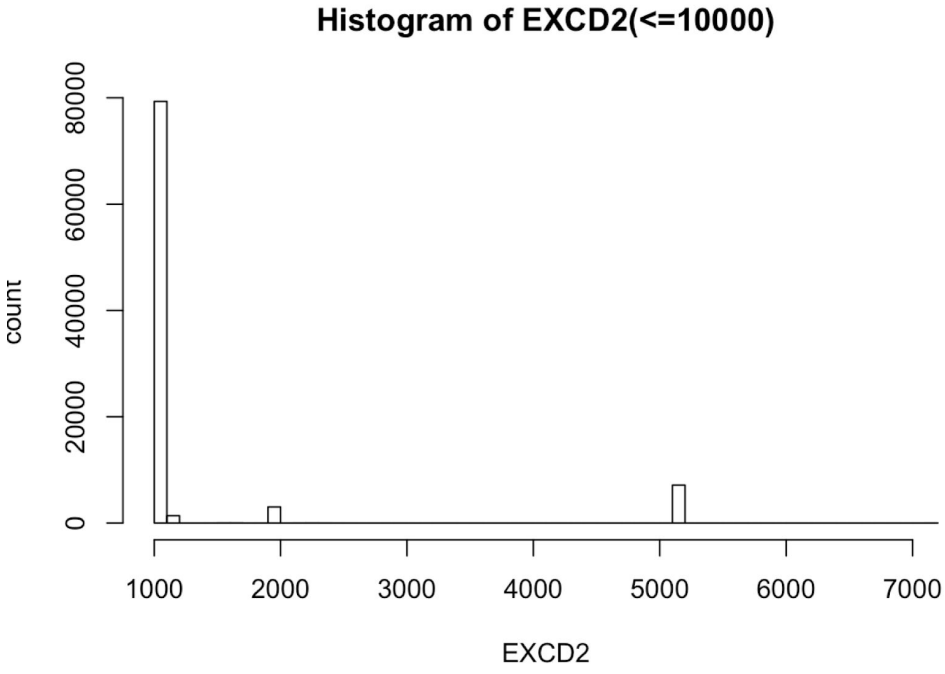
EXTOT2


Boxplot of EXTOT2
EXTOT2
5
10
15
20
0


Histogram of EXTOT2(<=100000)
count
0
10000
20000
30000
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
EXTOT2

EXCD2





STRING VARIABLES

STADDR (stands for standard address)

Top 10 addresses that have most properties in NY

	STADDR	total_count
1	501 SURF AVENUE	902
2	330 EAST 38 STREET	817
3	322 WEST 57 STREET	720
4	155 WEST 68 STREET	671
5	20 WEST 64 STREET	657
6	1 IRVING PLACE	650
7	*NA*	641
8	220 RIVERSIDE BOULEVARD	628
9	360 FURMAN STREET	599
10	200 EAST 66 STREET	585

OWNER

Top 10 owners that have most properties in NY

	OWNER	total_count
1	*NA*	31081
2	PARKCHESTER PRESERVAT	6021
3	PARKS AND RECREATION	3358
4	DCAS	2053
5	HOUSING PRESERVATION	1900
6	CITY OF NEW YORK	1189
7	NEW	1014
8	BOARD OF EDUCATION	1003
9	CNY/NYCTA	975
10	NYC HOUSING PARTNERSH	747

CITY OF NEW YORK

There are 197 different zip codes in this dataset. The following table shows the top 10 zip code.

	ZIP	total_count
1	*NA*	26356
2	10314	24605
3	11234	20001
4	10462	16905
5	10306	16576
6	11236	15678
7	11385	14921
8	11229	12793
9	11211	12710
10	10312	12634

BBLE

BBLE is concatenation of BORO, BLOCK, LOT, and EASEMENT. So we separate this variable to see the distribution of properties in NY's 5 region.

	BORO	BORO_name	total_count
1	4	QUEENS	357931
2	3	BROOKLYN	323221
3	1	MANHATTAN	146161
4	5	STATEN ISLAND	113642
5	2	BRONX	107180